Benchmarking Music Audio Pre-trained Representations

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Music and Technology

School of Music

Carnegie Mellon University, Pittsburgh, PA

Ruibin Yuan

Thesis Committee:
Roger B. Dannenberg, Chair
Gus Xia
Riccardo Schulz

May 2023

Acknowledgements

I would like to express my gratitude to Roger B. Dannenberg, Gus Xia, and Riccardo Schulz for their invaluable guidance and support throughout this research. Their expertise and insights have been instrumental in shaping the direction and scope of this study. I would also like to thank the MusicAudioPretrain Team for their support and encouragement. Their resources and tools have been immensely helpful in carrying out this research. And I would like to thank my love, Xinyue, for her support during my hard times. Finally, I would like to extend my heartfelt thanks to all the participants who generously gave their time and insights to make this study possible. Your contributions are deeply appreciated.

Abstract

Large-scale pre-trained models have demonstrated remarkable success in domains such as computer vision, natural language processing, and speech and audio processing. Despite this, the audio aspect of the music domain remains under-explored, and there is currently no benchmark to adequately assess the music-understanding capabilities of existing pre-trained music audio representations.

To address this knowledge gap, we introduce the **M**usic **A**udio **R**epresentation **B**enchmark for universa**L** **E**valuation, termed **MARBLE**. It aims to provide a benchmark for various Music Information Retrieval (MIR) tasks by defining a comprehensive taxonomy with four levels of hierarchy, including acoustic, performance, score and high-level description. We then establish a unified protocol based on 14 tasks on 8 public-available datasets in MARBLE, providing a fair, easy-to-use, extendable, and reproducible assessment of music audio representations, with a clear statement on copyright issues on datasets.

However, currently there are only a limited number of music representations available, and they come with various limitations. For instance, some only focus on music tagging, while others cannot perform sequence tasks. Furthermore, many of these representations lack large-scale pre-training, have insufficient parameter size, or are overly cumbersome.

In order to expand the quantity of music audio representation baselines and enhance music audio representation, we pre-train self-supervised learning methods used in speech. We evaluate them on the MARBLE benchmark and identify their limitations. Consequently, we proceed to design novel pretext tasks for music audio pre-training and propose a novel system called MERT. MERT achieves a balanced performance on MARBLE and the best MERT checkpoint matches the ensemble performance of previous state-of-the-art systems. However, there is still a large room for further improvement.

The leaderboard and toolkit repository are published[1] to promote future music AI research.

***K*eywords** Representation learning · Audio representation · Music representation · Music pre-training · Music information retrieval · Music understanding · Benchmark · Language model · HuBERT · Jukebox

[1] https://marble-bm.shef.ac.uk

Contents

List of Tables

List of Figures

Chapter 1

Introduction

AI is undergoing a paradigm shift with the rise of large-scale pre-trained models [8]. In recent years, pre-training has achieved significant results in domains like computer vision, natural language processing, and speech and audio processing. It can leverage large-scale unlabeled data to learn useful representations that benefit downstream resource-restricted tasks.

However, large-scale pre-training is not a well-studied topic in the music domain, especially the audio aspect of music, given the small scale of open-sourced music data and the lack of computing resources in this community. Also, even though a limited number of existing explorations like Jukebox [22, 10] and MusiCNN [56] have shown promising results, a universal benchmark to fully evaluate these models is missing. We do not know the pros and cons of each pre-training strategy.

To accelerate the paradigm shift in the music domain, this work aims to advance the research in this direction to **evaluate** and even **train** large-scale music audio pre-trained models. The resulting benchmark, the analysis and the proposed new systems can give insights into future music audio pre-training work. We focus on two main research questions, which lead into the research in chapter 2 to chapter 6, and will be answered in chapter 7:

- What have the existing music audio pre-trained representations learned?
- How can we design better pre-training strategies for music audio representation learning?

This work is a result of a big project in collaboration with a large team of which I am a core member. This thesis contains and reorganizes content from three papers of which I was a first or co-first author and which were produced by the project [40, 78, 46]. A detailed description of the work and division of labor will be presented in Chapter 8.

Chapter 2

Music Audio Representations

While lots of work on music representation is in the symbolic domain [16, 15, 80, 31, 73], less is in the audio. The existing pre-trained representations for audio and music are either trained in a supervised fashion or in a self-supervised fashion.

2.1 Supervised Audio Representations

With the building of the ImageNet project [19], lots of effective supervised methods for image classification have emerged. The most similar things to the Imagenet in the music domain are the Million Song Dataset (MSD) [4] and the Free Music Archive (FMA) [17]. Researchers in the MIR community have applied similar techniques to MSD and FMA, to pre-train music representations with supervision from the music tagging labels. A common practice is to use 1D or 2D CNN blocks to build a neural net that takes either the waveform or the spectrum as input and predicts the tagging labels [56, 14, 39, 34, 30]. However, as reported in [10], these methods do not work well on key estimation, a task that requires the knowledge of pitch, equal temperament, tonality, or even some level of music structure modeling. In fact, the key estimation accuracies of these methods are significantly worse than the handcrafted chroma baseline and similar to a classifier that only predicts the most prevalent class (F minor).

2.2 Self-supervised Audio Representations

Self-supervised learning (SSL) of data representations can be traced back to the days of word2vec [50] and has become more popular after the proposal of transformers [68] and BERT [20]. Currently, most of the popular methods in the audio area can be divided into three ways: reconstruction, contrastive learning, and mask prediction.

Reconstruction. The most intuitive way to model the data without supervision from labels is to simply reconstruct the data. Jukebox [22] is the most well-known one in this category. It tries to tackle the modeling of the extremely long audio sequences by gradually compressing it into lower dimensional discrete codebooks with auto-encoding. Then it approaches the generation by learning auto-regressive transformers on top of the low-resolution discrete codes. Apart from that, reconstructing handcrafted features will also benefit the resulting representation [75, 55, 59]. In general, reconstruction helps the network model more nuances of the data, and Jukebox representation achieves the best results on several MIR tasks among all the pre-trained representations [10]. However, reconstructing high-dimensional data can be expensive, and the evaluation of the reconstruction quality can be ambiguous sometimes.

Contrastive Learning. Another way is to model the consistency of the data. By introducing augmentations to the data, contrastive learning usually requires the network to learn a consistently similar representation of different views of the same sample and distinct representations of different samples. Some recent methods [66, 77, 61] follow this paradigm. These kinds of methods are usually sensitive to the quality of negative samples and suffer from representation collapse. They either use expensive sampling strategies for hard negative sample mining or use extremely large batch size to include as many negative samples as possible. Though there is an alternative way proposed recently [25, 54] to avoid the negative samples, the training can be unstable. Note that at least one of the methods in this category also suffers from bad key estimation accuracy.

Mask Prediction. Most of the pre-trained methods follow the paradigm of mask prediction. This paradigm leverages the distributional hypothesis, that data segments that occur in the same contexts tend to be semantically similar. In other words, masking a portion of the data and using the unmasked ones to predict the masked ones, can be an effective representation learning strategy. It is easier than reconstruction but also avoids the negative sample problem of contrastive learning. Some work [82, 83, 45, 64] attempts this by regressing the masked input features. More recently, state-of-the-art methods tend to perform mask prediction in a discrete space [28, 13]. They first construct a discrete codebook using VQVAE-ish methods or simple k-means, then train transformers to predict the masked discrete codes. It can be more effective in the sense that the quantization serves as a denoising procedure to help the learning focus more on the important concepts and semantics rather than modeling noise, thus making the learning easier. There is also work that combines the idea of contrastive learning with mask prediction [3, 2] and achieves promising results.

It is somewhat surprising that training on a dataset to accomplish some prediction and/or similarity task produces representations that are often useful in other tasks. The general idea is that prediction (reconstruction and mask prediction) and similarity (from contrastive learning) benefit from more abstract representations that capture (if we are lucky) general information. In the case of music, these representations might separate out more abstract properties such as pitch, harmony, texture, genre, etc. This training is less specific than task-oriented training, but it can benefit from huge amounts of unlabeled data.

Starting with the results of this training (which we now call “pre-training”) we can treat the learned abstract representation as an intermediate result, and we can train a second stage to take the representation as input and learn a specific task, i.e., a function from abstract representation to task-level representation such as pitch, genre, emotion, vocal technique, etc. These second stages usually require labeled data, which is limited. However, by starting from a good representation learned from a huge dataset, it is often the case that results are greatly improved when compared to trying to learn functions that take us directly from audio input to task-specific outputs.

Of course, pre-training does not guarantee success. For example, pre-training may not learn the “right” representations for certain tasks. We have seen this, for example, in pitch estimation tasks. Part of the research, then, is to develop pre-training strategies (such as introducing loss functions that directly bias systems to learn and encode pitch information) that facilitate a wide variety of downstream tasks. Hence, this work has two important components: MERT performs representation learning through pre-training on large amounts of audio data. MARBLE offers a suite of benchmarks to evaluate MERT using smaller datasets, well-specified tasks, and labeled data used for both training and evaluation. Improved training techniques for MERT aim to produce better evaluation scores in MARBLE. Extensions and improvements to MARBLE create challenges for MERT and suggest improved training strategies.

Chapter 3

MARBLE: Music Audio Representation Benchmark for Universal Evaluation

Due to issues such as copyright and annotation costs, labelled music datasets are usually small, which limits the performance of supervised models. Given that self-supervised learning (SSL) is versatile for various tasks (e.g., NLP [32, 26, 63] and CV [53]) with limited annotated datasets, there have been works on SSL-based audio representation learning [36, 45, 44, 3, 28, 61, 72] and music pre-trained models [56, 47, 82, 75, 42, 22, 40, 66, 77, 49]. The existing benchmarks, GLUE [70], SuperGLUE [69], and ERASER [21] in NLP, along with VTAB [81] and VISSL [24] in CV, all play an active role in promoting the development of SSL-related research topics in the corresponding domains. However, there are only scattered and fragmented evaluations of the existing music models rather than comprehensive benchmarks, and thus makes it difficult to objectively compare and draw insights across techniques.

In this chapter, we propose a Music Audio Representation Benchmark for universaL Evaluation (MARBLE) to address this problem. MARBLE aims to examine the full range of model capabilities, therefore proposes a taxonomy adapted from [16] to categorise MIR tasks, including acoustic, performance, score and high-level description. The four-level hierarchy aligned to musician consensus serves as a guideline to further organise the datasets and helps to identify a diversified set of downstream tasks. We select popular tasks in the (now defunct) Music Information Retrieval Evaluation eXchange (MIREX) Challenge[1], and use the corresponding public datasets with limited annotations. As demonstrated in Tab. 3.1, the current version of MARBLE contains 14 downstream tasks, spread over 10 task categories on 8 publicly or even commercially available datasets. Except for the common classification tasks, we also integrate the missing piece of the puzzle – sequence labelling tasks that require frame-wise prediction

[1] https://www.music-ir.org/mirex/wiki/MIREX_HOME

including source separation and beat tracking. The datasets used in MARBLE are ensured easy-to-access: all datasets are available for download directly from the official repository or an external website for downloading a specific version.

In addition, we design a unified protocol and build tool-kits in MARBLE to evaluate the generalisation ability of music representations. In the MARBLE protocol, the pre-trained systems will be regarded as backbones to provide universal representations for all tasks, and task-specific prediction heads are concatenated to further trained under *unconstrained*, *semi-constrained*, and *constrained* settings, which is defined by whether the training hyperparameters are restricted and whether the backbone model is frozen. The evaluation suite provides codes for dataset preprocessing and examples of evaluating existing popular systems in the benchmark.

3.1 Benchmark Tasks

As demonstrated in Tab. 3.1, we collect datasets in MARBLE to provide the community, with a standard, general-purpose, easy-to-use benchmark for a variety of tasks covering all aspects of music. Generally, music processing involves discriminative and generative tasks. The discriminative tasks either classify or regress musical recordings as a whole or use a seq2seq model to make frame-by-frame decisions on entire sequences. The generative tasks include audio synthesis and music composition. For the initial release of MARBLE, we focus on discriminative tasks, and generative tasks are currently outside our scope. The task collection is guided by the principles of (1) receiving a high level of interest in the MIR community, (2) having publicly available datasets allowing everyone to participate, and (3) limited labelled data to effectively measure the universality of the model. Four aspects of music are studied through 14 proposed tasks: **High-level description tasks** including key detection, music tagging, classification gender and emotion recognition; **Score-level tasks** including estimating the pitch of a musical note and tracking beats; **Performance-level tasks** including detecting musical ornaments or techniques; and **Acoustic-level tasks** including singer identification, instrument classification, and source separation that focus more on raw audio information.

3.1.1 High-level Description Tasks

Key detection involves predicting the scale and key pitch levels of a song. MARBLE solves this task using the Giantsteps [35] and a subset of the Giantsteps-MTG-keys dataset [37]. Giantsteps dataset contains 604 songs and is taken as our dedicated test set. Additionally, we leverage a subset of the Giantsteps-MTG-keys dataset, which contains 1077 music pieces with single-key annotations, for training and validation. Since

Table 3.1: The Dataset, Commercial License, and Prediction Head of Each Task Used for the MARBLE Benchmark. SDR refers Source-to-distortion Ratio.

Taxonomy	**Task Type**	**Task & Annotation**	**Prediction Type**	**Evaluation Metrics**	**Commercially Available**
High-level Description	Key Detection	Giantsteps key [35]	Multi-class	Weight score [57]	Yes
	Music Tagging	MagnaTagATune [38]	Multi-label	ROC-AUC & PR-AUC/AP	-
		MTG Top50 [7]	Multi-label	ROC-AUC & PR-AUC/AP	-
	Genre Classification	GTZAN [67]	Multi-class	Accuracy	-
		MTG Genre [7]	Multi-label	ROC-AUC & PR-AUC/AP	-
	Emotion Detection	Emomusic [65]	Regression	Emo_V & Emo_A	-
		MTG MoodTheme [7]	Multi-label	ROC-AUC & PR-AUC/AP	-
Score-level	Pitch Classification	Nsynth [23]	Multi-class	Accuracy	Yes
	Beat Tracking	GTZAN Rhythm [67]	Seq2Seq, Binary-class	F-measure (threshold 20ms)	-
Performance-level	Vocal Technique Detection	VocalSet [74]	Multi-class	Accuracy	Yes
Acoustic-level	Singer Identification	VocalSet [74]	Multi-class	Accuracy	Yes
	Instrument Classification	Nsynth [23]	Multi-class	Accuracy	Yes
		MTG Instrument [7]	Multi-label	ROC-AUC & PR-AUC/AP	-
	Source Separation	MUSDB18 [58]	Seq2Seq, Regression	SDR	-

no standardised split is available for Giantsteps-MTG, we adopt the dataset split strategy employed in [10]. Both datasets contain 2 minutes of electronic dance music covering all 12 pitch classes in major and minor, resulting in a 24-class classification task. For performance evaluation, we employ accuracy with an error tolerance metric, which is a weighted score metric. This metric grants partial credit for reasonable errors, such as predicting relative secondary keys when the primary key is the ground truth [57].

Music Tagging refers to assigning a predefined set of tags to a given song. These tags encompass various aspects such as genre, instrumentation, mood, and tempo (e.g., fast), making music tagging somewhat overlap with genre classification, emotion recognition, and instrument classification. To conduct our study, we utilise two extensive datasets: MagnaTagATune (MTT) [38] and MTG-Jamendo (MTG) [7]. The MTT dataset comprises 30-second audio clips with manual annotations for tags. It consists of 25.9k clips, amounting to a total duration of 170 hours. For MARBLE, we use the Top50 tags, and adopt a conventional (12:1:3) training, validation, and test split, aligning with all baseline approaches' practices. Besides, the MTG dataset contains 55k clips, corresponding to nearly 2k hours of music. As the audio clips in this dataset may exceed 30 seconds in length, we compute multiple embeddings using a sliding window of 30 seconds and then average them to obtain an overall embedding representation. While both datasets encompass a large number of tags, we follow the customary to limit the vocabulary to the 50 most common tags in each dataset. The evaluation metrics employed for this task are the macro-average of all tag ROC-AUCs (receiver operating characteristic - area under the curve) and the average precision (AP) / PR-AUC (precision-recall - area under the curve). These metrics provide comprehensive insights into the model's performance across all tags.

Genre classification aims to assign each song the most suitable genre label. This study uses two distinct datasets: GTZAN [67] and MTG-Genre. GTZAN consists of 30-second audio clips from 10 genres, making it suitable for a multi-class classification task. To assess the performance of this dataset, we report the accuracy metric. To ensure consistent evaluation, we utilise the "fail-filtered" split as described in [33] for GTZAN. The filtered dataset comprises 930 audio tracks corresponding to approximately 8 hours of music. Besides, MTG-Genre, derived from MTG-Jamendo, contains 55k tracks but focuses solely on 95 genre tags, resulting in a multi-label classification problem. We employ the ROC and AP metrics to evaluate the performance on MTG-Genre.

Emotion Recognition in music aims to determine the emotional content of music pieces. In our study, we utilise two distinct datasets to evaluate the performance of emotion recognition: Emomusic [65] and MTG-MoodTheme [7]. Emomusic contains 744 pieces of 45-second music clips and is annotated with valence and arousal scores. The valence represents the positivity of emotional responses, while arousal indicates emotional intensity. The official evaluation metric for this dataset is the determination of the coefficient (r^2) between the model's regression results and human annotations of arousal and valence [65]. During inference, we split the 45-second clips into 5-second sliding windows and computed the average prediction probability as the final prediction. Since no standard dataset split is available for Emomusic, we adopt the same partitioning as [10]. It is important to note that direct comparison of the SoTA model's results with the benchmark may be challenging due to the different dataset splits. Additionally, we utilise MTG-MoodTheme, a subset of MTG-Jamendo consisting of 18.5k audio tracks annotated with 59 human emotion labels. This is a multi-label task with ROC and AP as evaluation metrics.

3.1.2 Score-level Tasks

Pitch Classification in Music (Monophonic) involves determining the appropriate pitch category for a given audio sample, ranging from MIDI note numbers 0 to 127 on a semitone scale. In this study, we perform pitch classification using the Nsynth dataset [23] within the music information retrieval benchmark. It comprises 340 hours of music, with each excerpt lasting 4 seconds. Since the audio recordings in this dataset are monophonic, the pitch classification task is formulated as a 128-class classification problem, covering all possible MIDI pitch categories (fundamental frequencies from 8Hz to 12.5kHz). The evaluation metric used for this task is the accuracy achieved across all audio clips.

Beat Tracking involves determining the presence of a beat and a downbeat in each frame of a given music piece. In this benchmark, we only focus on beat tracking, making it a binary-classification task[2].

[2] Due to the limitation of time and the size of the dataset, tracking the time signature (e.g., 4/4 metre) and downbeat is deferred to future versions with other datasets.

An offline approach is employed for beat tracking, allowing the model to utilise frame-level information during inference. The model generates frame-by-frame predictions at a specific frequency, which are then post-processed using a dynamic Bayesian network (DBN) [6] implemented with `madmom` [5] to obtain the final result. The GTZAN Rhythm dataset [48] is used in this study. The dataset provides frame-level annotations for each music clip in GTZAN. To enhance model performance and ensure a fair comparison with other popular systems, adjacent frames of each beat label are also labelled as beats using a label smoothing technique commonly employed in beat tracking. The model is evaluated using the `f_measure` metric implemented in `mir_eval` [57]. A prediction is considered correct if the difference between the predicted event and the ground truth does not exceed 20ms. It is important to note that while some models may have been trained on other datasets, the GTZAN-train subset is used as the training set, and GTZAN-test is used as the test set for all MARBLE submissions.

3.1.3 Performance-level Tasks

Vocal Technique Detection task involves identifying different singing techniques within an audio clip. For this task, the MARBLE benchmark utilises the VocalSet dataset [74], the sole publicly available dataset specifically designed for studying singing techniques. This dataset comprises recordings of 20 professional singers (9 female and 11 male) performing 17 distinct singing techniques in various contexts, amounting to a total duration of 10.1 hours. Given that the audio clips are segmented into 3-second intervals, the task focuses on determining the type of technique (e.g.Vibrato, Straight) rather than the precise start and end times. To evaluate the performance of models, we employ Accuracy as the evaluation metric. We use a subset of 10 different singing techniques used in [76], which contains 15 singers in training and validation set, and 5 for the test set. Since there is no predetermined division between the training and validation sets, we assign 9 singers to the training set and 6 singers to the validation set. It is important to note that all 3-second segments originate from the same audio recording file within the same part of the split, such as being exclusively part of the training set. Detailed data partitioning can be found in our provided code.

3.1.4 Acoustic-level Tasks

Instrument Classification refers to the multi-label or multi-class identification of instruments present in a given audio recording. In the MARBLE benchmark, we utilise two datasets: Nsynth and MTG-instrument. The Nsynth dataset comprises 306,000 audio tracks, with each track corresponding to one of 11 different instruments. The evaluation metric for this dataset is accuracy. On the other hand, MTG-instrument is a subset of MTG-Jamendo, containing 25,000 audio tracks and 41 instrument tags. Each track can have multiple instrument tags and is evaluated based on ROC and AP.

Singer Identification involves recognizing the singer or vocal performer from an audio recording. In previous work on Singer Identification using the VocalSet dataset [74], different splits are employed. For the MARBLE benchmark, we randomly split the dataset into training, validation, and test sets, maintaining a ratio of 12:8:5. All sets contain the same 20 singers. The specific data divisions can be found in the provided code.

Source Separation aims to separate different components of a music recording, such as vocals, drums, bass, and others. In MARBLE, we adopt the widely-used MUSDB18 dataset [58] for this task. MUSDB18 consists of 150 full-length music tracks, totaling approximately 10 hours of audio and multiple isolated stems. Our training set consists of 86 tracks, the validation set contains 14 tracks, and the evaluation set comprise 50 tracks, following the official MUSDB18 setting. During training, we randomly sample 6-second segments and apply random track mixing for data augmentation. Due to the complexity of this task, we utilise the baseline architecture from the Music Demixing Challenge (MDX) 2021 [51]. This architecture consists of three linear layers and three bi-directional LSTM layers. The optimization is performed by directly computing the l2-loss between the predicted and ground-truth linear magnitude spectrograms. The evaluation metric for this task is the Source-to-Distortion Ratio (SDR) as defined in [51], which is calculated as the mean across the SDR scores of all songs.

3.2 Downstreams and Training Strategies

To evaluate the relevance of representations for downstream MIR tasks, we design evaluation frameworks: the *unconstrained* track, *semi-constrained* track and the *constrained* track. In the unconstrained track, researchers are invited to submit their systems with any hyperparameter and structure configuration, including the option to fine-tune pre-trained models. This track encourages flexibility and exploration, enabling researchers to investigate a wide range of approaches. On the other hand, the semi-constrained track requires the submissions to use frozen pre-trained backbones. Finally, the constrained track employs a standardised setting with limited hyper-parameter search space, where frozen models are used as feature extractors for training a one-layer 512-unit MLP (or 3-layer 512-unit LSTM for source separation) on each task. The hyper-parameter search space is defined as below:

1. **Layer**: {every single layer, weighted sum}
2. **Model**: {one-layer 512-units MLP, 3-layer 512-unit LSTM (source separation only)}
3. **Batch size**: {64}

4. **Learning rate**: {5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}

5. **Dropout probability**: {0.2}

In addition, we set a computational wall for MARBLE. The systems need to finish each task within a week on our machine equipped with a single consumer GPU (RTX3090). By offering these three evaluation tracks, we aim to provide researchers with a comprehensive platform to assess the performance and relevance of representations in MIR tasks, encouraging innovative approaches and fostering advancements in the field.

For the same task with a uniform dataset, if there are different evaluation metrics (e.g., emotion regression, source separation, and tagging), we will average the two evaluation metrics. We select the checkpoints regarding to the best validation results for final testing and submission.

The experiments and results are discussed in chapter 6.

Chapter 4

Speech Self-supervised Learning on Music

In order to expand the quantity of music audio representation baselines and enhance music audio representation, in this chapter, we pre-train two speech-based self-supervised learning (SSL) methods on music. We evaluate them on the MARBLE benchmark, learn some lessons through the ablation studies, and identify their limitations.

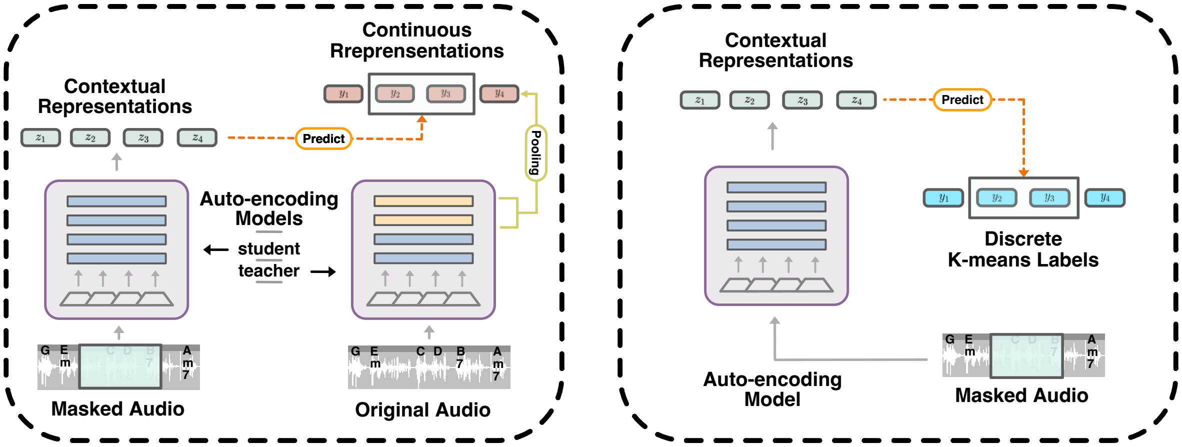
4.1 Speech SSL Frameworks Applied to Music

In this section, we briefly describe the two selected SSL models – data2vec-1.0 [2] and Hubert [28] – in the unified auto-encoding framework (cf. Fig. 4.1) and discuss the similarities and differences under music audio pre-training.

4.1.1 Music2Vec: Continuous Target Prediction

We adapt the pre-training paradigm from the speech version of the multi-modal framework data2vec-1.0 [2], where the prediction targets during pre-training are continuous representations. We refer to this continuous prediction model adapted with music recordings as Music2Vec.

Modified from the design of bootstrap your own latent (BYOL) [25], Music2Vec aims to predict continuous latent representations from the teacher model for the masked input audios, which is illustrated in Fig. 4.1(a). The teacher model and student model share the same architecture, and the parameters of the teacher model are updated according to the exponential moving average of the student [2]. The student model takes the partially masked input and is asked to predict the average pooling of top-K layer outputs from the Transformer [68] in the teacher model. In contrast, the teacher model takes the unmasked input and provides contextual prediction targets in the pre-training.



((a)) Music2Vec ((b)) MusicHuBERT

Figure 4.1: Pre-training Paradigms of Music2Vec and Music HuBERT. Both of the models are fed with masked audio inputs and predict given targets without supervised information.

4.1.2 MusicHuBERT: Discrete Target Prediction

In contrast, another efficient speech self-supervised model, HuBERT [28], is chosen as the representative of discrete target prediction design. We referred to the music adaption version of the model as MusicHuBERT.

The MusicHuBERT model takes masked music audio as input (Similar to Music2Vec) and predicts pre-processed discrete labels corresponding to the masked area, as shown in Fig. 4.1(b). The discrete targets are pseudo labels provided by K-means that are trained on the MFCC features of the training audios. The number of clusters K of the K-means model is a hyperparameter, and all the centroids are assigned with randomly initialised embeddings and learned during the MusicHuBERT pre-training. MusicHuBERT can also be trained for an extra n iterations, where the K-means clustering is learned from model outputs' previous iteration. We follow the original speech HuBERT [28] setting to train a model with 95 million parameters of the same size as Music2Vec.

4.1.3 Design Comparision

Although both Music2Vec and MusicHuBERT are annotation-free and utilize self-supervised learning, their most common characteristic is the training task of "reconstructing" information from masked inputs, making them auto-encoding models. During the denoising process, these models learn the semantics contained in the audio. Furthermore, they share similar model architecture designs, which are inherited from wav2vec-2.0 [3], wherein the audio is initially encoded by a multi-layer 1-D CNN feature extractor that maps a 16 kHz waveform to 50 Hz representations. The encoded tokens are then fed into a 12-layer Transformer Block with a hidden dimension of $H = 768$ (with $4 \times H$ feed-forward inner-dimension).

Regarding the differences in the designs, the most notable one is that Music2Vec is required to predict continuous latent variables, whereas MusicHuBERT predicts discrete pseudo-labels. The time cost of the SSL target preparation bottleneck varies according to the mechanism. In Music2Vec, the pre-training consumes twice the model forward time since the target representations from the teacher model are inferred on-the-fly. In contrast, MusicHuBERT trains the K-means model and infers all the pseudo-labels before training, which requires high parallel processing ability when the dataset is scaled-up.

4.2 Pre-training on Speech versus Music

The pretraining dataset we used is a private dataset, including 1000 hours of music audio recordings; each sample is a 30s-long excerpt from pop-song or instrumental music. The size of the pre-training dataset is roughly the same as the pre-training for HuBERT-base and data2vec-audio-base models. The Facebook fairseq framework[1] is used for the pre-training. We termed the music variants of HuBERT-base and data2vec-audio-base as MusicHuBERT (MuHuBERT) and Music2Vec. All the MuHuBERT and Music2Vec models are trained for 400k steps with 8 × NVIDIA A100-40GB GPUs. Training with eight GPUs takes around 2 days, i.e., about 20 days with only one A100 GPU.

Table 4.1: Experimental performance of the SSL baseline systems on all downstream tasks

Downstream dataset	MTT		GS key	GTZAN Genre	EMO		Nsynth Instr	Nsynth pitch	VocalSet tech	VocalSet singer	GTzAN Rhythm	MTG Instrument		MTG MoodTheme		MTG Genre		MTG Top50	
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	ROC	AP	ROC	AP	ROC	AP	ROC	AP
Hubert base	89.8	36.4	15.0	64.8	31.0	57.5	68.2	79.4	61.0	58.8	83.5	73.2	17.0	74.0	11.6	85.0	16.3	81.8	26.5
MuHubert base	**90.2**	**37.7**	14.7	**70.0**	**42.1**	**66.5**	69.3	77.4	65.9	**75.3**	**88.6**	**75.5**	**17.8**	**76.0**	**13.9**	**86.5**	**18.0**	**82.4**	**28.1**
data2vec audio base	88.4	33.6	15.5	60.7	23.0	49.6	69.3	77.7	64.9	74.6	36.4	73.1	16.9	73.3	11.0	83.5	14.5	80.6	24.8
Music2vec vanilla	89.1	35.1	**19.0**	59.7	38.5	61.9	**69.4**	**88.9**	**68.3**	69.5	33.5	73.1	16.3	74.3	12.2	85.2	16.5	81.4	26.2
SOTA	**92.0**[30]	**41.4**[10]	**74.3**[37]	**82.1**[39]	**61.7**	**72.1**[10]	**78.2**[72]	**89.2**[49]	65.6[76]	**80.3**[52]	80.6[27]	**78.8**	**20.2**[1]	**78.6**	**16.1**[49]	**87.7**	**20.3**[1]	**84.3**	**32.1**[49]

Table 4.1 demonstrates the performance of HuBERT[2] and data2vec[3] SSL models that were pre-trained on speech recordings and music recordings separately. Here, we only consider the SOTA performance trained with the same dataset train/valid/test split. All of the models are used as parameter-frozen deep feature extractors. The weighted sum of all layers representations are evaluated under the MARBLE setting.

For the HuBERT model, the results pre-trained on music audio are comparable with state-of-the-art (SOTA) and surpass the HuBERT pre-trained on speech audio with exceptions on pitch estimation on Nsynth and key detection on GS. For data2vec, we can observe similar trend with exceptions on beat tracking on GTZAN-Rhythm and singer identification on Vocalset.

[1] https://github.com/facebookresearch/fairseq
[2] https://huggingface.co/facebook/hubert-base-ls960
[3] https://huggingface.co/facebook/data2vec-audio-base

We can tell that MusicHuBERT is more promising than Music2vec given that it provides better results in most of the downstream tasks, especially genre classification on GTZAN, emotion regression on EMO and beat tracking on GTZAN. But it is worse on single-pitch estimation on Nsynth, along with key detection on GS.

4.3 Ablation Studies

Here, we carry out an ablation study of hyperparameter search under both pre-training paradigms. Given the time limitation, we did not extract features on MTG datasets and only calculated the results in another 9 downstream tasks.

4.3.1 Ablation Study on MusicHuBERT

Table 4.2: Ablation study on MusicHuBERT hyperparameters (k is the number of MFCC clusters)

Downstream dataset	MTT		GS key	GTZAN Genre	EMO		Nsynth Instr	Nsynth pitch	VocalSet tech	VocalSet singer	GTZAN Rhythm	Average Score
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	score
Hubert	89.8	36.4	15.0	64.8	31.0	57.5	68.2	79.4	61.0	58.8	83.5	59.8
k=2000 MFCC dim=39	90.2	37.7	14.7	**70.0**	42.1	66.5	69.3	77.4	65.9	75.3	88.6	64.4
k=2000 iter2	**90.4**	37.5	13.8	68.3	**43.3**	67.4	70.0	**80.3**	63.6	70.4	**88.8**	63.8
k=500 MFCC dim=39	89.6	36.1	15.7	64.5	41.0	67.7	66.7	76.8	60.5	72.3	87.5	62.4
k=500 MFCC dim=60	90.3	**38.0**	**17.6**	69.7	40.8	**67.5**	**70.3**	79.0	**66.2**	**75.5**	88.6	**65.0**

We use the number of clusters k =500 and k=2000. For the case k=500, we increase the dimension of MFCC features from 13, which is commonly used in the speech community, to 20, which is widely used in sound event detection. Thus, the dimension of MFCCs combined with their delta features and delta-delta features have 39 and 60 dimensions respectively. For the case of k=2000, we use the 768-dimension deep feature learned from the first iteration experiment to carry out the second iteration k-means.

From Table 4.2, we can see that MusicHuBERT with k=2000 is better than the k=500 case for most of the tasks. Given HuBERT is good for speech when k=100 or k=500, which is roughly the number of human phonemes, this implies music tokens or notes are much richer than speech and therefore need a larger number for quantisation.

The results on k-means for deep features are better than the vanilla MusicHuBERT besides genre classification on GTZAN, singer identification on vocalset, and singing techniques classification on vocalset. This implies the MFCCs features are good for modelling the human voice, regardless of speech or singing. The results of GTZAN may be due to the randomness as the dataset is very small.

Besides, increasing the dimension of MFCCs provides no significant difference among most of the tasks other than key detection and both tasks on Nsynth. Increased dimensionality for MFCC features can provide more detailed information on impulse response for a sound event. Thus, monophonic instrumental notes can be better modelled with 60-dimension MFCCs features. Furthermore, the emotion regression also provides different results, but the average of the two metrics are nearly the same, providing no significant improvement.

4.3.2 Ablation Study on Music2Vec

Table 4.3: Ablation study on Music2Vec hyperparameters (span is mask span, prob is mask probability, step is training steps, target=12 uses all 12 transformer layers, and crop5s uses 5s music excerpts)

Downstream	MTT		GS	GTZAN	EMO		Nsynth	Nsynth	VocalSet	VocalSet	GTZAN	Average
dataset			key	Genre			Instr	pitch	tech	singer	Rhythm	Score
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	score
data2vec	88.4	33.6	15.5	60.7	23.0	49.6	69.3	77.7	64.9	**74.6**	**36.4**	55.2
vanilla	89.1	35.1	19.0	59.7	38.5	61.9	69.4	88.9	68.3	69.5	33.5	57.8
span=5	87.3	32.0	15.7	47.6	22.7	41.2	64.2	84.8	56.7	53.8	33.2	49.7
span=15	88.7	34.3	16.4	56.6	39.0	58.8	67.1	88.1	63.1	61.9	33.1	55.2
prob=50	88.5	34.0	23.7	59.3	40.6	55.0	66.8	87.7	64.9	61.7	33.9	56.3
prob=80	88.2	33.9	18.4	50.3	36.7	55.7	67.9	88.9	64.2	65.2	33.7	55.1
step=800k	87.7	32.7	20.3	54.5	34.9	47.3	66.9	87.5	65.6	65.1	33.4	55.0
target=12	89.7	35.2	**26.5**	64.5	41.7	64.2	**71.1**	**89.2**	71.0	73.2	34.1	60.6
crop5s	**90.0**	**36.6**	18.5	**76.6**	**53.4**	**71.6**	68.3	88.9	**71.3**	72.4	33.9	**61.8**

We use audio files with 30s length, mask span length 10, mask probability 65%, target top-8 transformer layer the teacher model as a deep feature, and training step 400K as the vanilla setting. We conduct parameter searching and correlation analysis for Music2Vec pretraining, including the masking strategy, training steps, the learning target layers, and recording length; the results are shown in Table 4.3.

We revise the masking strategy by changing the **mask span length** and **mask token probability** in the data2vec-audio-base setting. Mask token probability is the probability for each token to be chosen as the start of the span to be masked, the length of which can also be adapted for different data modalities. The results in Table 4.3 show that the other span value and other mask token probability provide a bit worse results on nearly all the tasks. This suggests that the data2vec hyperparameters for speech pre-training are generally helpful for music pre-training.

Given the fact that early transformer layer representations generally perform well on key detection and beat tracking, we modify the **prediction target** provided by the teacher model. We change the prediction target in Music2Vec from the original one, that is, the average of the top-8 layer representations, to all the 12 layers. The results in Table 4.3 show that Music2Vec actually benefits, not only from the potentially preserved key information shown by a significant increase on GS but all the other tasks as well.

Furthermore, we use **audio length cropping** to shorten music excerpts since longer sequences are more difficult to model. Note that the combined audio length in a batch on a single GPU is not altered, and the hardware environment remains the same, making a single training batch contain a larger number of music samples when clips are cropped.

4.4 Limitations of Directly Applying Speech SSL to Music

There is still a large gap between the musically pre-trained speech SSL and the SOTA on every downstream task, especially the performance on GS and EMO. Furthermore, if we select MusicHuBERT as the approach for scaling up, since it delivers more promising results on most downstream tasks, the pitch estimation accuracy will also be a problem.

Although Music2Vec has shown promising results in some tasks, its overall performance is mediocre. We suspect that it may be limited by the instability of negative contrastive learning, which often leads to representation collapse. We have attempted to train multiple trails of music2vec, but the results were not consistent. Table 4.1 presents results from the best checkpoint.

We posit that in the HuBERT paradigm, the lower MFCC coefficients, we believe, convey general spectral shape such as formants (important for speech) and don't model individual harmonics or their frequencies (or pitch). Also, the employment of k-means clustering on MFCC features to obtain latent classes is likely to capture a greater extent of information at the acoustic level, while not being particularly efficacious in capturing information pertaining to pitch and tonal levels. Therefore, the lack of multi-pitch information in the k-means teacher resulting in a sub-optimal modelling of pitch, harmony and key, that is essential to GS and NSynth.

In the next chapter, we will propose a new approach to mitigate the limitations of the aforementioned speech SSL methods.

Chapter 5

MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training

In the previous chapter, we have shown that directly applying speech SSL methods to music may result in a sub-optimal learning paradigm. This is primarily due to the distinctive challenges associated with modelling musical knowledge, particularly tonal and pitched characteristics of music. To address this research gap, in this chapter, we propose an acoustic **M**usic und**ER**standing model with large-scale self-supervised **T**raining (**MERT**), which incorporates teacher models to provide pseudo labels in the masked language modelling (MLM) style acoustic pre-training.

In our exploration, we identify a superior combination of teacher models, which outperforms conventional speech and audio approaches in terms of performance. This combination includes an acoustic teacher based on Residual Vector Quantization - Variational AutoEncoder (RVQ-VAE) and a musical teacher based on the Constant-Q Transform (CQT). These teachers effectively guide our student model, a BERT-style transformer encoder, to better model music audio. See Fig. 5.1.

Furthermore, we explore a wide range of settings to overcome the instability in acoustic language model pre-training, which allows our designed paradigm to scale from 95M to 330M parameters.

Experimental results indicate that our model can generalise and perform well on 14 music understanding tasks. Compared to existing state-of-the-art methods, our model performs at approximately the same level on average, and improves on the state of the art in some tasks.

Note that the results in this chapter do not follow the constrained settings of MARBLE, as we directly cite the baseline numbers reported by previous work. We will re-evaluate the baseline systems as well as the proposed MERTs with the MARBLE constrained protocol in the next chapter.

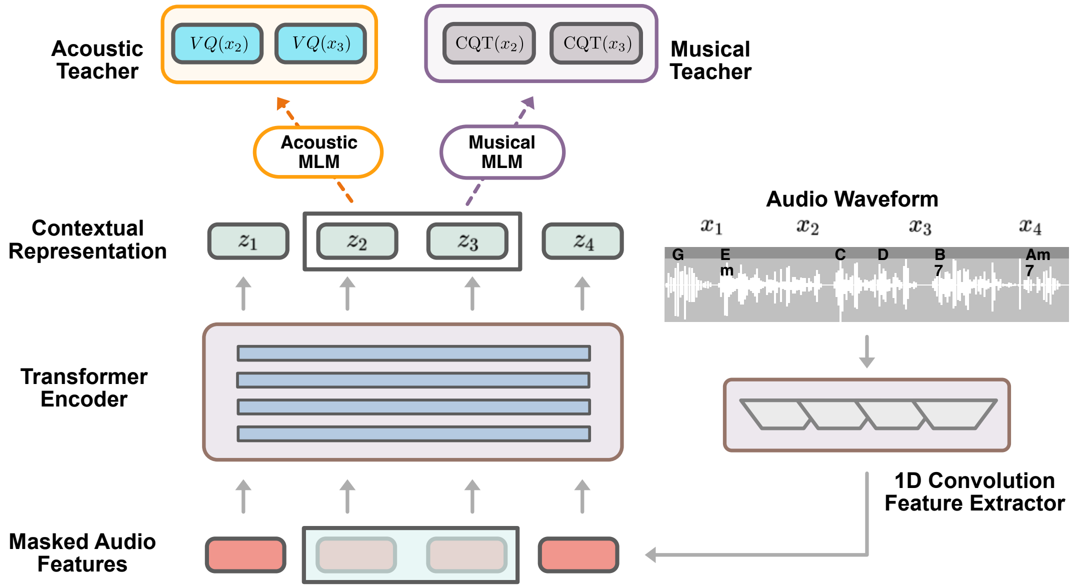


Figure 5.1: Illustration of the MERT Pre-training Framework.

5.1 Methodology

This section introduces the pre-training paradigm and architecture of our models. It includes prediction to acoustic teachers such as k-means or deep music features, and reconstruction to music teachers such as CQT spectrum, both based on the well-established masked language model (MLM) paradigm.

5.1.1 Pre-training with MLM

Supervised Learning requires a labelled dataset $\mathcal{D}_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N}$. Here, N is the number of data samples, $x_i^{(t)}$ is the i^{th} data sample in the dataset, and $y_i^{(t)}$ is the corresponding label. From $\mathcal{D}_t$, we can train a machine learning algorithm $f_\theta(\cdot)$ parameterised with θ that makes label predictions on each data sample. Unsupervised learning, in contrast, learns an algorithm based on an unlabelled dataset $\mathcal{D} = \{x_i\}_{i=1}^{M}$, with SSL being a specific type of this class. For each data sample x_i, SSL derives a new data x_i' with a pseudo label y_i'. The training process is to minimise the loss between each pseudo label y_i' and the prediction based on new data $\hat{y}_i = f_\theta(x_i')$ as denoted in Eq.5.1.

$$\theta^* = arg\,min_\theta \sum_{x_i^{(t)} \in D} \mathcal{L}\left(f_\theta(x_i'^{(t)}), y_i'^{(t)}\right). \tag{5.1}$$

MLM is a famous example of pseudo-label generation. Let $x_i = \left[x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(L)}\right]$ be the i^{th} data sample in a speech or language dataset with length L, and $M \subset [L]$ is a subset of indices randomly chosen from 1 to L. Then, the new data is defined by the following equation

$$x'_i = \left[\mathbf{1}_{[L]\setminus M}(1) \cdot x_i^{(1)}, \mathbf{1}_{[L]\setminus M}(2) \cdot x_i^{(2)}, \cdots, \mathbf{1}_{[L]\setminus M}(L) \cdot x_i^{(L)}\right] \tag{5.2}$$

where $\mathbf{1}_{[L]\setminus M}(x)$ denotes the indicator function, that is, $\mathbf{1}_{[L]\setminus M}(x) = 1$ if and only if x is outside the masked indices set M. The pseudo-label that needs to be learned is typically $y'_i = x_i - x'_i$, i.e., the masked data. However, reconstructing masked data y' for raw audio tasks as pseudo-label is hard to train. HuBERT [68, 28] uses a dimension-reduced feature z' derived from y' with phonetic acoustic information, which forms the design basis of our pre-training strategy.

As a speech SSL system, HuBERT utilises offline clustering to acquire pseudo labels for a BERT-like prediction loss. Specifically, it uses Mel-frequency cepstral coefficients (MFCCs), a widely-used traditional feature in speech-related tasks, as acoustic features for clustering. The obtained results are then utilised as pseudo labels in the first iteration of pre-training. It then uses the learned representation for clustering to get a pseudo label for the second iteration pre-training. Such a pseudo label includes acoustic information in human speech and can be aligned to phonemes. The loss functions of HuBERT are formulated as follows:

$$\mathcal{L}_H(f; x, M, Z) = \sum_{t \in M} \log p_f(z_t \mid x', t) \tag{5.3}$$

where $\log p_f(\cdot \mid x', t)$ is the log-likelihood function on clustering results given the masked input x' and position t derived from f; likelihood function p_f is the Noise Contrastive Estimation (NCE) loss which is defined as

$$p_f(c \mid x', t) = \frac{\exp(\mathrm{sim}(T(o_t), e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(T(o_t), e_{c'})/\tau)}, \tag{5.4}$$

Here, $c \in [C]$ is a codeword of the clustering results and e_c represents its embedding; sim is the cosine similarity; o_t is the output of the model at timestep t; and $T(o_t)$ is the linear transformation of o_t, making it have the same dimension as e_c. Besides, τ scales the logit and is set to 0.1 in HuBERT. The linear transformation T, the model to generate outputs, and the embedding of all the clustering results are all learnable.

Overall, we use the same model as HuBERT but introduce several notable variations tailored to music. Specifically, we designed a better hidden-unit z as pseudo tags for pre-training with multiple music acoustic features. In addition, we added a reconstruction loss to music features and employed additional music augmentation tricks.

5.1.2 Modelling Acoustic Information

The MFCC features are only good at modelling acoustic and timbre information for single-pitch signals, and therefore, the clustering results do not provide much timbre information in music recording. We proposed two potential approaches as the teacher on acoustic information: one based on traditional features, and the other based on deep learning.

The first method uses k-means on the log-Mel spectrum and Chroma features for timbre and harmonic acoustic information, respectively. In the case of music representation, each frame contains more information compared to speech, necessitating a larger number of classes for k-means clustering. The complexity of the k-means algorithm is linear with the number of centroids (clustering centers), leading to a time-consuming k-means for the music feature. To tackle this problem, we employ 300-means for the log-Mel spectrum with dimension 229, and 200-means for Chroma features with dimension 264, resulting in a total of 60,000 classes (200 centroids for Chroma features multiplied by 300 centroids for the log-Mel spectrum). Despite the increased number of classes, the computational complexity remains comparable to that of HuBERT. The disadvantage of k-means is that it is difficult to scale up to a larger number of classes and larger datasets, and the results are sensitive to initialisation.

The second choice for our acoustic teacher is EnCodec [18], a recent learnable feature with 8-layer residual Vector Quantized-Variational AutoEncoder (VQ-VAE). Each acoustic feature, denoted as $z_{enc} \in [C]^{L\times 8}$, is a 2-dimensional auditory code matrix, and L is the length of the recording. The row vector of each matrix $z_{enc}[t,:]$ represents the results of 8 different clusterings for frame t, and the column vector of each matrix $z_{enc}[:,j]$ represents the results from the j^{th} codebook of the audio sequence, where $j \in \{1,\ldots,8\}$. EnCodec converts 24kHz input waveforms to 8 different embeddings at 75Hz with a 320-fold reduction, and the quantizer has 1024 dimensions. In this setting, for each 5-second waveform, the discrete acoustic feature is a matrix with 375 × 8 entries, representing 375 frames (75Hz × 5s) and 8 deep acoustic features. With these embeddings, the decoder of EnCodec can reconstruct the waveform at 24 kHz with authentic information in timbre.

5.1.3 Modelling Musical Information

Apart from acoustic information, we added a new reconstruction loss to the Constant-Q transform (CQT) spectrogram to emphasise pitch-level information. The CQT is a type of frequency transform that is widely used in various MIR tasks, such as pitch detection, chord recognition, and music transcription. It is similar to the Fourier transform, but bin widths are proportional to frequency rather than equal, giving each octave the same number of bins, resulting in a better time-frequency trade-off for music audio where multiple pitches occur in multiple octaves. We utilize mean squared error (MSE) loss to reconstruct the CQT spectrum z_{cqt} from the masked input audio x'. That is,

$$\mathcal{L}_{CQT}(f_{cqt}; x, M, \mathbf{z}_{cqt}) = \sum_{t\in[L]} \left\| z_{cqt,t} - f_{cqt}(x')_t \right\|_2 \tag{5.5}$$

And the final loss function $\mathcal{L}$ is a linear combination of both the acoustic loss function $\mathcal{L}_H$ and the musical-pitch loss function $\mathcal{L}_{CQT}$:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_H + \mathcal{L}_{CQT} \tag{5.6}$$

5.1.4 Robust Representation Learning

We introduce "in-batch noise mixup" for music SSL. The mixup augmentation refers to the audio clip being mixed with a certain ratio of shorter audio excerpts to form an augmented single sample during pre-training, instead of using the original audio. We randomly sample the audio segments from the same batch and add them to audio at random positions according to some probability. Theoretically, sampling from the whole training dataset would provide more randomness and thus be more beneficial to the representation robustness, but we narrow the sampling pool to the same audio batch considering the limited computational resources. The mixup could enable the learning of more robust musical representations and force the model to focus on the useful musical source and to ignore the noise. A pseudocode implementation can be found in Appendix A.

5.2 Training Settings

We deploy the proposed SSL architecture in the training of various model sizes with matched scales of data. We mined 160K hours of music recordings from the Internet to build a large-scale music dataset. Accordingly, the base (95M) size models are trained with a 1K hours subset whereas the whole dataset is used for the large (330M) model. Specifically, we provide a special edition of the base model, `MERT-95M-public`, that is trained on a totally publicly available music dataset, music4all [62], with a data size of 910 hours. In the context of self-attention, the computational complexity scales quadratically with the sequence length. Therefore, when dealing with limited computational resources, there exists a trade-off between the batch size and the sequence length. In our preliminary experiments, we have observed that increasing the batch size provides greater performance improvements compared to extending the context length. To ensure manageable computation during pre-training, we adopt a strategy of randomly truncating audio clips into 5-second segments. This duration roughly corresponds to a 2-bar context in music. It is worth noting that our model utilizes a convolutional relative positional embedding, similar to the approach introduced by [3] in Wav2Vec, enabling it to operate effectively in longer contexts if required. The effective batch sizes and learning rates for the base model and large model are set to 1.5 and 5.5 hours, and their learning rates are set to 5e$-$4, 1.5e$-$3 respectively. Pre-training of our models has been carried out with the fairseq[1] framework. The base and large models are trained with 64 A100-40GB GPUs with half-precision settings.

5.3 Training Stability

In our empirical findings, we have observed that when scaling up acoustic encoder-only models, they tend to exhibit a higher susceptibility to training instability compared to models of similar size in natural language processing and computer vision domains. Such instability can result in decreased performance or, in extreme cases, even lead to crashes in model training. During our experimentation with scaling up to the `MERT-330M` model, we encountered notable instability manifested by constant gradient clipping and sporadic spikes in losses. This instability had a detrimental effect on the accuracy of masked language modeling (MLM) predictions and resulted in decreased performance on downstream tasks. Our attempts to resume training from previously saved checkpoints and data batches proved unsuccessful in mitigating the instability. Furthermore, we observed that reducing the learning rate in this context not only failed to address the issue but also led to a decline in performance and hindered the convergence of the model. We further explored the effectiveness of a seemingly-powerful method DeepNorm [71] in stabilizing acoustic language model pre-training but found it to be ineffective in this particular scenario. Additionally, we discovered that incorporating attention relaxation techniques [12] proved beneficial in addressing the instability challenges we encountered. However, we found that transitioning from post-layer normalization (Post-LN) to pre-layer normalization (Pre-LN) offered a potential solution by alleviating the instability and allowing training to continue. More information can be found in appendix A.0.1.

5.4 Performance & Efficiency of MERT Models

The results on all the downstream tasks are provided in Tab. 5.1 and Tab. 5.2. As suggested by the average scores in Tab. 5.2, we averaged best published scores on these tasks including those achieved from supervised methods. Not only does $\texttt{MERT-330M}^{\texttt{RVQ-VAE}}$ achieve the same average, but it exceeds previous performance on 4 of the metrics. It is also noteworthy that the other smaller `MERT-95Ms` still have close performance when using fewer parameters. Generally, MERT models perform well on tasks focusing on local-level musical information such as beat, pitch and local timbre such as singer information, and remain competitive on the other tasks such as music tagging, key detection, and genre classification, which require more global-level information. This indicates the blending of acoustic and musical teachers could provide comprehensive guidance for the understanding of music recordings, though pre-trained in only a 5-second

[1] https://github.com/facebookresearch/fairseq

Table 5.1: Experimental Performances of MERT and Baselines on Downstream Tasks (1/2). The baselines are grouped by supervised and unsupervised pre-training paradigms. The superscripts denote the category of the acoustic teacher used by MERT models. "public" refers to the MERT model trained with only open-source dataset. Results with star* are claimed in the references.

Dataset	**MTT**		**GS**	**GTZAN**	**GTZAN**	**EMO**		**Nsynth**		**VocalSet**	**VocalSet**
Task	Tagging		Key	Genre	Rhythm	Emotion		Instrument	Pitch	Tech	Singer
Metrics	ROC	AP	Acc$^{\text{Refined}}$	Acc	F1$^{\text{beat}}$	R2$^{\text{V}}$	R2$^{\text{A}}$	Acc	Acc	Acc	Acc
MusiCNN [56]	90.6*	38.3*	12.8*	79.0*	-	46.6*	70.3*	72.6	64.1	70.3	57.0
CLMR [66]	89.4*	36.1*	14.9*	68.6*	-	45.8*	67.8*	68.3	47.0	60.0	50.7
Jukebox-5B [11, 79]	91.5*	**41.4***	66.7*	79.7*	-	**61.7***	72.1*	70.0	90.9	77.6	81.4
MULE [49]	91.4*	40.4*	66.7*	73.5*	-	57.7*	70*	74.0*	89.2*	75.5	**87.8**
HuBERT-base$^{\text{music}}$ [28]	90.2	37.7	14.7	70.0	**88.6**	42.1	66.5	69.3	77.4	65.9	75.3
data2vec-base$^{\text{music}}$ [2]	89.1	35.1	19.0	59.7	33.5	38.5	61.9	69.4	88.9	68.3	69.5
MERT-95M$^{\text{K-means}}$	90.6	38.4	65.0	78.6	88.3	53.1	68.7	71.3	91.5	74.6	77.2
MERT-95M-public$^{\text{K-means}}$	90.7	38.4	66	71.4	88.1	53.2	71.5	69	91.1	75.5	78.2
MERT-95M$^{\text{RVQ-VAE}}$	91	39.3	63.3	78.6	88.3	60	76.4	69	91.7	74.2	83.7
MERT-330M$^{\text{RVQ-VAE}}$	91.3	40.2	65.6	79.3	87.9	61.2	**74.7**	71.3	**92.4**	**78.3**	87.3
Previous SOTA	**92.0** [29]	**41.4** [11]	**74.3 [37]**	**83.5** [49]	80.6 [27]	**61.7**	72.1 [11]	**78.2** [72]	89.2 [49]	65.6 [76]	80.3 [52]

Table 5.2: Experimental Performances of MERT and Baselines on Downstream Tasks (2/2). Average scores across *task* are calculated on the SOTA results and models applicable to all the tasks.

Dataset	**MTG**		**MTG**		**MTG**		**MTG**		**MUSDB**				
Task	Instrument		MoodTheme		Genre		Top50		Source Seperation				**Avg.**
Metrics	ROC	AP	ROC	AP	ROC	AP	ROC	AP	SDR$^{\text{vocals}}$	SDR$^{\text{drums}}$	SDR$^{\text{bass}}$	SDR$^{\text{other}}$	
MusiCNN [56]	76.2	18.6	74.7	12.8	86.0	17.5	82.0	27.3	-	-	-	-	-
CLMR [66]	73.5	17.0	73.5	12.6	84.6	16.2	81.3	26.4	-	-	-	-	-
Jukebox-5B [11, 79]	-	-	-	-	-	-	-	-	5.1*	4.9*	4.1*	2.7*	-
MULE [49]	76.7	19.6	78.1	15.3	88	20.4	83.7	30.7	-	-	-	-	-
HuBERT-base$^{\text{music}}$ [28]	75.5	17.8	76.0	13.9	86.5	18.0	82.4	28.1	4.7	3.7	1.8	2.1	55.8
data2vec-base$^{\text{music}}$ [2]	73.1	16.3	74.3	12.2	85.2	16.5	81.4	26.2	5.5	5.5	4.1	3.0	51.3
MERT-95M$^{\text{K-means}}$	77.2	19.6	75.8	13.6	87.0	18.6	82.8	29.4	5.6	5.6	4.0	3.0	62.8
MERT-95M-public$^{\text{K-means}}$	76.9	19.2	76.2	13.4	87.2	18.9	82.9	28.7	5.5	5.5	3.7	3.0	62.3
MERT-95M$^{\text{RVQ-VAE}}$	76.5	19.2	76.5	13.6	87.0	18.8	82.7	28.3	5.5	5.5	3.8	3.1	63.4
MERT-330M$^{\text{RVQ-VAE}}$	77.0	19.7	76.7	14	87.0	18.6	83.5	29.7	5.3	5.6	3.6	3.0	**64.5**
Previous SOTA	**78.8**	**20.2** [1]	**78.6**	**16.1** [49]	**87.7**	**20.3** [1]	**84.3**	**32.1** [49]	**9.3**	**10.8**	**10.4**	**6.4** [60]	**64.5**

context length. Nevertheless, the performance of our models in tasks with more global music information are close to state-of-the-art, suggesting MERT models are capable of recognising global patterns well, thanks to the use of relative position embeddings and the contextualisation of the transformer network. Further work can be focused on modelling longer context.

In addition, our model can demonstrate good results with limited data, and public data may lack enough diversity. For one thing, `MERT-95M-public` and `MERT-95M` are both trained on a ∼1k hour dataset. Both of them have comparable results with the SOTA and `MERT-330M`, proving that MERT can converge effectively and learns generalisable music representations with limited training data. For another, the `MERT-95M-public` is trained with Music4ALL [62], a 910-hours public music dataset with mainly pop music and lack of diversity in music style. The experimental results show comparable performance to other settings. In particular, its performance does not have a significant difference besides genre classification on GTZAN compared to `MERT-95M`. This suggests our model can acquire a powerful representation even with a dataset that is not representative.

Moreover, we evaluated the performance of the $\texttt{MERT}^{\texttt{RVQ-VAE}}$ model with a parameter size of 95M and 330M, given the use of the EnCodec feature enables us to scale up the model compared to the k-means feature. The results demonstrate that increasing the model size to 330M yields improved performance or has a very small difference (less than 0.1%) in performance on most of the tasks besides beat tracking.

More importantly, the lightweight sizes of MERTs open up new possibilities for transferring one general understanding model for large-scale classification or sequence labelling MIR tasks. MERT series models achieve better or comparable performance with only 1.9% (95M) and 6.6% (330M) parameters compared to the self-supervised baseline Jukebox-5B [22]. Even when our evaluation is in probing setting, most models could not be trained on sequence labelling tasks like beat tracking or source separation with affordable computational costs except for MERT and baseline models with similar architecture [28, 2].

Table 5.3: Evaluation Results from Models Trained with Different Teacher Settings. Models labeled with $^{\triangle 2}$ and $^{\blacktriangle 2}$ suggest that the K-means teachers are trained with the features from $^{\triangle 1}$ and $^{\blacktriangle 1}$ models. All the listed models are in base size (95M) and not augmented with the in-batch noise mixture.

Acoustic Teacher	**Acoustic Target Class**	**Musical Teacher**	**MTT** Tagging		**GS** Key	**GTZAN** Genre	**EMO** Emotion		**Avg.**
			ROC	AP	Acc^{Refined}	Acc	R2^{V}	R2^{A}	
K-means^{MFCC}	100	N/A	89.8	36.3	15.1	66.2	39.6	67	49.4
K-means^{MFCC}	500		90.3	38	17	70	40.6	67.5	51.3
K-means^{MFCC}	2000$^{\triangle 1}$		90.2	37.6	15.6	70	44.3	67.6	51.4
K-means$^{Logmel+Chroma}$	300 + 200 $^{\blacktriangle 1}$		90.5	37.6	55.1	75.2	40.1	68.2	62.1
K-means^{MFCC}	2000$^{\triangle 2}$		90.4	37.5	16.1	68.3	43.9	67.7	51.0
K-means$^{Logmel+Chroma}$	500$^{\blacktriangle 2}$		90.4	37.7	49.2	72.8	46.5	66.9	60.7
K-means$^{Logmel+Chroma}$	300 + 200	CQT	**90.6**	**38.4**	65.0	**78.6**	53.1	68.7	**67.3**
RVQ-VAE	1024×8 $^{all\ codebook}$	CQT	90.5	**38.4**	63.2	77.2	**53.2**	**72.3**	66.9
	1024 ^{codebook7}		88.6	34.4	63.5	62.1	33.3	53.2	57.6
	1024 ^{codebook0}		90	36.7	59.4	67.2	39.7	64.5	60.5
	1024×8 $^{random\ codebook}$		**90.6**	38.1	**66.8**	73.8	48.1	68.6	65.8

5.5 The Effectiveness of Acoustic & Musical Teacher

As demonstrated in Tab. 5.3, we explore optimal combinations and selections of the teacher models in the MERT paradigm with a subset of downstream tasks, including auto-tagging (MTT), key detection (GS), genre classification (GTZAN), and emotion recognition(EMO).

We reproduce the original HuBERT [28] setting on music datasets with only the acoustic teacher K-means$^{MFCC\triangle 1}$ and the teacher K-means$^{MFCC\triangle 2}$ trained on features produced by HuBERT model from the first stage similar to DeepCluster [9]. We observe that such models perform poorly on the key detection and emotion recognition tasks even we increase the dimension of the MFCC features from 100 to 2000. As the re-clustering K-means does not bring significant improvement in the second stage pre-training, we stick to the ordinary one stage pre-training to study the influence brought by the teachers with less computational cost.

Given that the key information is highly related to the pitch classes of the audio, we then introduce such inductive bias by providing the K-means acoustic teacher with both Logmel and Chroma features, denoted as K-means$^{\text{Logmel+Chroma}\blacktriangle 1}$. The additional pitch information indirectly brought by the Chroma feature immediately endows the model a certain of level of key detection ability and raises the accuracy from 15.6 to 55.1 while keeping or increasing performances on other tasks. This confirms that the potentials of transformer models can be better excavated from more dimensions by introducing extra pseudo prediction targets in the MLM scheme.

Following such an intuition, it could be further assumed that designing a proper multi-task learning pre-training paradigm can guide the model to produce more general representations for various music understanding tasks. We thus propose leveraging the CQT musical teacher to introduce harmonic inductive bias during the pre-training. Compared to models trained with only the acoustic teacher $^{\text{MFCC}\triangle 1}$ or K-means$^{\text{Logmel+Chroma}\blacktriangle 1}$, MERT models trained with the newly proposed CQT musical teacher that are naturally more aligned to music audio can achieve significant performance gains on not only the key detection task but also the tasks requiring the high-level information like genre classification and emotion recognition.

However, given that K-means models are difficult to scale up on large-scale datasets due to the memory and computational requirements, we use the RVQ-VAE model EnCodec [18] as the final version of our acoustic teacher without looking for the immeasurable hyper-parameter K for music audio. The EnCodec could intuitively provide more comprehensive acoustic information since the audio can be largely recovered from the acoustic prediction targets, i.e. the intermediate discrete codecs produced by the EnCodec encoder, by a neural decoder from the RVQ-VAE.

We observe that leveraging only one top ($1024^{\text{codebook7}}$) or bottom layer ($1024^{\text{codebook0}}$) of the residual codebooks in RVQ-VAE can provide abundant information in pre-training, the utilisation of all layers in the codebooks allows the student models to learn more sufficient acoustic patterns. While the strategy of randomly accessing one of the codebooks for each batch can alleviate the use of GPU memory and lead to similar performance compared to using all of them at a time, the setting of predicting 8 coodebooks all together is adopted in the final version of MERT and further utilised in the 330M scaling-up pre-training due to faster convergence. By replacing the acoustic teacher with RVQ-VAE, MERT can achieve average score 66.9 similar to 67.3 from the K-means$^{\text{Logmel+Chroma}\blacktriangle 1}$ version while leaving the possibility of scaling up with more training data.

5.6 Ablation Study on Loss Weight & Mixup Probability

We conducted a hyperparameter search to determine the optimal weight for the musical loss applied to masked audios in the k-means setting. Additionally, we investigated the impact of in-batch noise mixup

Table 5.4: Evaluation Results for Pre-training Setting Ablation Study.

Parameter Size	**Acoustic Teacher Model**	**Acoustic Target Class**	**Musical Loss Weight**	**In-batch Mixup Probability**	**MTT** Tagging		**GS** Key	**GTZAN** Genre	**EMO** Emotion		**Avg.**
					ROC	AP	Acc$^{\text{Refined}}$	Acc	R2$^{\text{V}}$	R2$^{\text{A}}$	
95M	K-means$^{\text{Logmel+Chroma}}$	300 + 200	N/A	N/A	90.5	37.6	55.1	75.2	40.1	68.2	62.1
			1	N/A	90.6	38.4	65.0	**78.6**	53.1	68.7	67.3
			2	N/A	90.6	38.1	62.7	66.9	45.5	67.9	62.7
			5	N/A	90.4	37.3	65.3	70.3	45.7	68.3	64.1
			1	0.25	90.6	37.9	**65.5**	70.0	49.6	72.5	65.2
			1	0.5	90.7	38.6	64.9	72.8	45.3	71.9	65.2
95M		1024×8 $^{\text{all codebook}}$	1	N/A	90.5	38.4	63.2	77.2	53.2	72.3	66.9
95M	RVQ-VAE	1024×8 $^{\text{all codebook}}$	1	0.5	**91.0**	**39.3**	63.3	**78.6**	**60.0**	**76.4**	**68.8**

augmentation on each training sample. We applied the same weight and mixup probability for both the EnCodec setting and the large model setting. In Table 5.4, we present the results of our pre-training setting ablation study, which uses the same evaluation setting in § 5.5. The table includes various parameters and evaluation metrics for different acoustic teacher models and target classes.

We further explored the influence of different musical loss weights for the 95M K-means model with Logmel and Chroma features. By adjusting the musical loss weight, we observed a decrease in performance on all of four tasks and found that a weight of 1 yielded the best performance for the base model. Additionally, we alter the in-batch mixup probability to evaluate whether it is affecting the performance of the model. We found the mixup probability provides worse results in MERT$^{\texttt{K-means}}$ but provides better performance for MERT$^{\texttt{RVQ-VAE}}$. Therefore, we determined a probability of 0.5 to be suitable based on the average performance score. Such a phenomenon deserves more attention.

Overall, our ablation study provides valuable insights into the impact of different settings and parameters on the performance of the audio language model. These findings can inform the development of more effective and efficient models in the domain of audio language processing.

Chapter 6

Re-Evaluation & Discussions

In this chapter, we re-evaluate all the baselines including the proposed ones under the same comparable setting of the MARBLE constrained protocol. We will discuss what we have learned from the evaluation results.

6.1 Baseline Representations

The music pre-trained representations selected for evaluation are summarised in Table. 6.1, in total 9 different versions of 7 pre-trained representations. Note that we do not cover models designed entirely for speech or not open source models.

Table 6.1: Information of Baseline Systems.

Method	**MusiCNN**	**CLMR**	**Jukebox**	**MULE**	**MAP-Music2Vec**	**MAP-MERT-v0**		**MAP-MERT-v1**	
	MSD-big					**base**	**base-public**	**base**	**large**
Network	CNN	9-Conv	3-Conv, 36-Trans	22-Conv, 2-Trans	7-Conv, 12-Trans	7-Conv, 12-Trans	7-Conv, 12-Trans	7-Conv, 12-Trans	7-Conv, 12-Trans
#Params	8M	2.5M	5B	62.4M	95M	95M	95M	95M	330M
Input	log-mel	waveform	waveform	log-mel	waveform	waveform	waveform	waveform	waveform
Stride	3s	2.69s	23.78s	2s	20ms	20ms	20ms	13.3ms	13.3ms
Context Length	3s	2.69s	23.78s	3s	30s	5s	5s	5s	5s
Data (hour)	10~20k	1.7k	60~120k	117.5k	1k	1k	0.9k	17k	160k
Pre-training Task	Music Tagging	Contrastive Learning	CALM	Contrastive Learning	MLM Boostrapping	MLM Clustering	MLM Clustering	MLM Clustering	MLM Clustering

MusiCNN [56] is a convolutional model pre-trained on the music audio tagging task using the MSD dataset [4]. We use the default configuration of the method, which is to concatenate the mean pooling of the CNN features for a 3-second input with the output of the maximum pool.

Contrastive learning of musical representations (CLMR) [66] leverages a 9-layer 1-D convolutional kernel as the feature extractor, employing a number of data augmentation, and is trained on both MSD and MTT. Both are trained with a contrastive learning approach. The model extracts an embedding every 2.69 seconds.

Jukebox [22] is a music generation model trained using codified audio language modelling (CALM). It is trained on 1.2 million private songs, and it is difficult to estimate the exact number of hours. However, assuming an average song length of 3-6 minutes, the total length could be 60k-120k hours, which is large and diverse to allow Jukebox to learn patterns and structures of different musical genres and styles. We use the same mid-layer representation as [10] to improve computational efficiency. Unlike other representations that run on short context windows, JUKEBOX is trained on a long window of 8192 sample points (23.78 seconds) of audio. We use the same strategy as [10] to extract the audio features on the downstream dataset.

MULE (Musicnet-ULarge) [49] is a SSL system based on **SF NFNet-F0** [72], SlowFast Normalizer-Free ResNet. It combines a SlowFast (SF) part (including a slower pathway that captures spatial information and a faster pathway that captures temporal information) with a more efficient and scalable variant of the Normalizer-Free ResNet (NFNet). MULE is contrastively pre-trained on the whole MusicSet dataset [49] and provides promising results on classification tasks. The model extracts an embedding with a 3-second window length and a 2-second hop length.

Music2Vec is a self-supervised learning (SSL) model is a system proposed in chapter 4. It is based on a bootstrapping mask prediction pre-training strategy. It consists of two main components: the student and teacher models. Both share the same architecture with 12 transformer layers, with the teacher model's parameters being exponential moving averages of the student model's parameters. The student model takes in masked input, and during training, it aims to learn deep features from the teacher model based on the output of the unmasked input. Specifically, it computes the average of the top 8 layers of the Transformer's output in the teacher model. To train the Music2Vec model, a private dataset comprising approximately 1,000 hours of music data was used. The input length of the Music2Vec model is set to 30 seconds, and it produces 50 embeddings per second. These embeddings capture essential features of the music data and can be utilised for various downstream tasks, including sequential tasks such as source separation and beat tracking.

MERT-v0, also referred to as MERT-95M$^{\text{K-means}}$ is a system proposed in chapter 5. It is a pre-trained model built upon the speech self-supervised learning (SSL) system HUBERT [28]. It undergoes pre-training for masked prediction, with discrete pseudo-labels obtained from K-Means clustering on music features. The pre-training task of MERT-v0 involves two pseudo-labels based on logmel and Chroma, along with a CQT reconstruction task that emphasises pitch information. Two versions of the MERT-v0 model are included: MERT-v0[1], trained on a private dataset of 1,000 hours, and MERT-v0-public[2], trained on Music4ALL [62]. The input length of the MERT-v0 model is set to 5 seconds, generating 50 embeddings per second. This design facilitates fine-tuning for sequential tasks, enabling efficient and effective processing of music data.

MERT-v1 encompasses two variants: MERT-v1-base[3] and MERT-v1-large[4]. They are proposed in chapter 5. These models, also known as MERT-95M$^{\text{RVQ-VAE}}$ and MERT-330M$^{\text{RVQ-VAE}}$, employ EnCodec, a pre-trained discrete deep feature, as a replacement for the K-means feature. This modification facilitates the scaling up of the model. Similar to MERT-v0, the input length of the MERT-v1 models is 5 seconds, but there are 75 embeddings per second. This configuration enables effective fine-tuning for sequential tasks, making the models suitable for processing music data in a variety of applications.

6.2 Results and Discussion

Table 6.2: Performance of Baselines Evaluated on MARBLE with constrained settings (1/2). We include previous SOTAs for reference. Note that MARBLE imposes strict constraints on downstream structures and hyper-parameter search spaces, while previous SOTAs are not subject to such limitations. Best scores on MARBLE are **bolded**, and best scores among all systems are underlined.

Dataset	**MTT**		**GS**	**GTZAN**	**GTZAN**	**EMO**		**Nsynth**	**Nsynth**	**VocalSet**	**VocalSet**
Task	Tagging		Key	Genre	Rhythm	Emotion		Instrument	Pitch	Tech	Singer
Metrics	ROC	AP	Acc$^{\text{Refined}}$	Acc	F1$^{\text{beat}}$	R2$^{\text{V}}$	R2$^{\text{A}}$	Acc	Acc	Acc	Acc
MusiCNN [56]	90.3	37.8	14.4	73.5	-	44.4	68.8	72.6	64.1	70.3	57.0
CLMR [66]	89.5	36.0	14.8	65.2	-	44.4	70.3	67.9	47.0	58.1	49.9
Jukebox-5B [10, 79]	**91.4**	**40.6**	63.8	**77.9**	-	57.0	73.0	70.4	91.6	76.7	82.6
MULE [49]	91.2	40.1	64.9	75.5	-	**60.7**	73.1	**74.6**	88.5	75.5	**87.5**
Music2Vec [42]	90.0	36.2	50.6	74.1	68.2	52.1	71.0	69.3	93.1	71.1	81.4
MERT-v0-95M [41]	90.7	38.2	64.1	74.8	**88.3**	52.9	69.9	70.4	92.3	73.6	77.0
MERT-v0-95M-public [41]	90.7	38.4	**67.3**	72.8	88.1	59.1	72.8	70.4	92.3	75.6	78.0
MERT-v1-95M [40]	91.0	39.3	63.5	74.8	**88.3**	55.5	**76.3**	70.7	92.6	74.2	83.7
MERT-v1-330M [40]	91.1	39.5	61.7	77.6	87.9	59.0	75.8	72.6	**94.4**	**76.9**	87.1
Previous SOTA	92.0 [29]	41.4 [10]	74.3 [37]	83.5 [49]	80.6 [27]	61.7	72.1 [10]	78.2 [72]	89.2 [49]	65.6 [76]	80.3 [52]

According to Table 6.2, 6.3 and Fig 6.1, all pre-trained baseline representations on MARBLE have achieved respectable results. Despite strict constraints on downstream structures and hyper-parameter search spaces, they are able to approach, if not surpass, the previous state of the art (SOTA) in many tasks. For instance, the best performance on NSynth Pitch classification have achieved up to 94.4% accuracy. Nonetheless, the majority of tasks are still far from being solved, including music tagging and source separation tasks. Notably, the performance on MUSDB18 is merely half of the previous SOTAs.

Our proposed models achieve balanced results, successfully performing tasks including sequence labelling, which other baselines fail to accomplish (as they do not provide frame-level representations or

[1] https://huggingface.co/m-a-p/MERT-v0

[2] https://huggingface.co/m-a-p/MERT-v0-public

[3] https://huggingface.co/m-a-p/MERT-v1-95M

[4] https://huggingface.co/m-a-p/MERT-v1-330M

Table 6.3: Performance of Baselines Evaluated on MARBLE with constrained settings (2/2). The overall average scores are calculated on the systems applicable to all tasks. Note that we denote the scores of *Jukebox-5B* on *MTG* tasks with asterisks(*), because it hit the computational wall of MARBLE, meaning that the system was unable to complete the corresponding task within a week on our machine equipped with a single consumer GPU (RTX3090).

Dataset **Task**	**MTG** Instrument		**MTG** MoodTheme		**MTG** Genre		**MTG** Top50		**MUSDB** Source Separation				**Avg.**
Metrics	ROC	AP	ROC	AP	ROC	AP	ROC	AP	$\text{SDR}^{\text{vocals}}$	$\text{SDR}^{\text{drums}}$	SDR^{bass}	$\text{SDR}^{\text{other}}$	
MusiCNN [56]	74.0	17.2	74.0	12.6	86.0	17.5	82.0	27.5	-	-	-	-	-
CLMR [66]	73.5	17.0	73.5	12.6	84.6	16.2	81.3	26.4	-	-	-	-	-
Jukebox-5B [10, 79]	**78.5***	**22.0***	77.6*	15.3*	**88.0***	**20.5***	83.7*	30.6*	-	-	-	-	-
MULE [49]	76.6	19.2	**78.0**	**15.4**	**88.0**	**20.4**	**83.7**	**30.6**	-	-	-	-	-
Music2Vec [42]	76.1	19.2	76.7	14.3	87.1	18.8	83.0	29.2	5.5	5.5	**4.1**	3.0	59.9
MERT-v0-95M [41]	76.6	18.7	75.9	13.7	86.9	18.5	82.8	28.8	**5.6**	**5.6**	4.0	3.0	62.3
MERT-v0-95M-public [41]	77.5	19.6	76.2	13.3	87.2	18.8	83.0	28.9	5.5	5.5	3.7	3.0	63.0
MERT-v1-95M [40]	77.5	19.4	76.4	13.4	87.1	18.8	83.0	29.0	5.5	5.5	3.8	**3.1**	63.3
MERT-v1-330M [40]	78.1	19.8	76.5	14.0	86.7	18.6	83.4	29.9	5.3	**5.6**	3.6	3.0	**64.2**
Previous SOTA	78.8	20.2 [1]	78.6	16.1 [49]	87.7	20.3 [1]	84.3	32.1 [49]	9.3	10.8	10.4	6.4 [60]	64.5

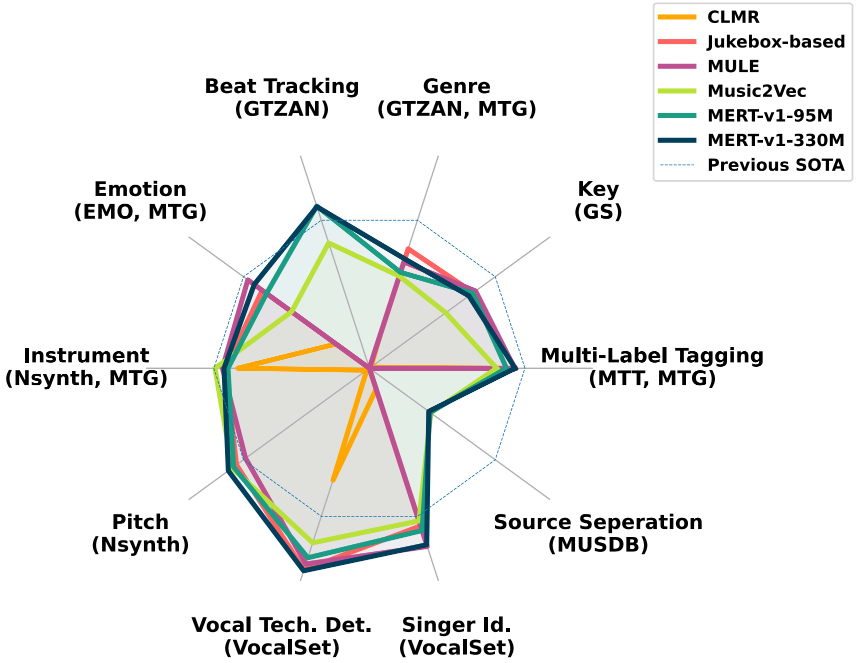
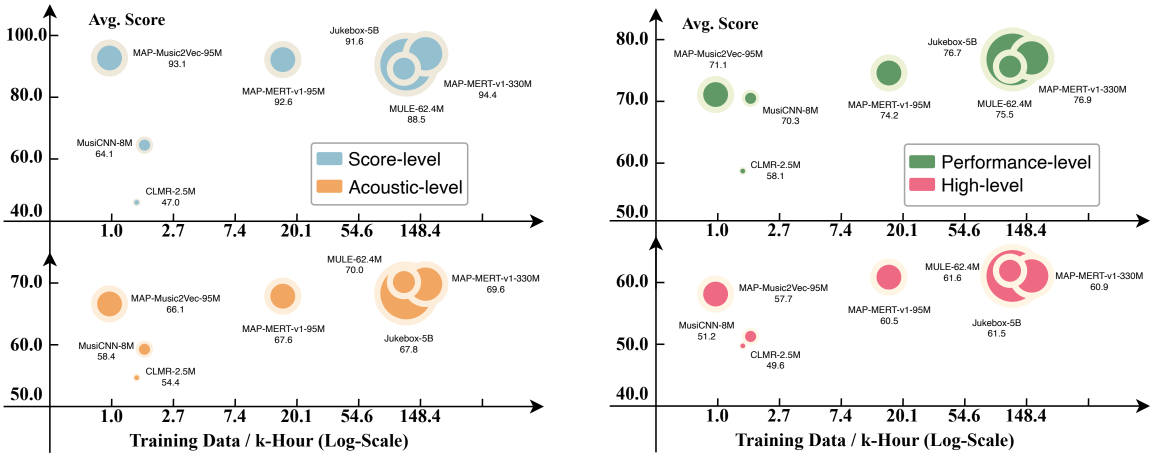


Figure 6.1: SSL Baselines Compared to previous SOTA. The performances of the tasks are merged according to the task types demonstrated in Tab. 3.1. Results not applicable are set to 0.

are too cumbersome to train). This series of models excel at multiple taxonomy levels. On certain tasks, MERTs achieve results close to or surpass the previous state-of-the-art. However, music tagging tasks are dominated by Jukebox-5B and MULE. Jukebox may benefit from its massive parameter size and generative modelling of detailed information, as well as the introduction of metadata during the pre-training period. Conversely, MULE benefits from its proprietary large-scale, high-quality dataset, MusicSet, and the highly discriminative representations learned by contrastive pre-training.

Based on Fig. 6.2(a) and 6.2(b), excluding sequence labelling tasks (as some baselines do not support them), we observe a general trend: as the volume of data and the size of model parameters increase, the performance of tasks across four levels correspondingly improves. The choice of pre-training method



((a)) Scores at Acoustic-level and Score-level.

((b)) Scores at Performance-level and High-level.

Figure 6.2: Results Analysis Regarding to Training Data Size. Since some models are not applicable to the sequence labelling tasks, the performances of *source separation* and *beat tracking* tasks are excluded on acoustic-level and score-level average score calculation correspondingly. The radii of the scatter points are isometrically log scaling with the parameter sizes.

and model size significantly influences the performance. For instance, Music2Vec-95M, utilizing only 1k hours of data for self-supervised learning, outperforms both supervised pre-trained MusiCNN-8M and contrastive pre-trained CLMR-2.5M on the same scale of data.

We will summarize our findings and answer the two research questions in the next chapter.

Chapter 7

Conclusions

In addressing the under-explored domain of music audio in large-scale pre-trained models, we introduced the Music Audio Representation Benchmark for universaL Evaluation (MARBLE). This provides a comprehensive, four-level taxonomy for Music Information Retrieval (MIR) tasks, as well as a unified protocol for assessing music audio representations through 14 tasks across 8 public datasets. We enhanced currently limited and flawed music representations by pre-training self-supervised learning methods used in speech and evaluated their constraints via MARBLE. This led to the creation of our system, MERT, which achieves balanced performance on the benchmark and matches the previous state-of-the-art systems' ensemble performance, though there remains significant scope for further advancements. We now return to the two main questions that were raised in the introduction and that motivated this research. With the extensive studies, we can begin to form some answers:

What have the (existing or proposed) music audio pre-trained representations learned? All the representations have learned multiple levels of knowledge. They are particularly good at high-level music descriptions, such as genre and emotion. However, when pre-trained with full supervision, the representations may not be able to model pitch and key information well, as they overfit to the supervision signal. SSL methods usually mitigate this issue by providing more generalizable representations. Many representations do not support frame-level representations and thus fail to run on sequence tasks like source-separation and beat tracking.

How can we design better pre-training strategies for music audio representation learning? A good pre-training strategy needs to be self-supervised in order to prevent overfitting to the supervision signal. And the method should be able to scale up to larger data and model size. Larger data and model size affects the performance more than the training paradigm (generative, contrastive or mask prediction) at this stage of research. Stacked transformer models are good candidates for future pre-training architecture, as they can be easily scaled up, and usually provide frame-level representations (this depends on the design though). However, how to stablize the training when scaling up remains an open question. Besides, we have also shown that, using pitch and tonal teachers to distill music knowledge into the representation is beneficial.

The findings in this work is still limited. We have a few insights about music audio pre-trained representations, but still need a lot of work to better understand the mechanisim behind the learning process and the training stability. For example, future work should investigate more downstream tasks and expand the MARBLE benchmark, do interpretable analysis, and address the training instability when scaling up the audio transformers (e.g. by removing the CNN tokenizer).

[10] Rodrigo Castellon, Chris Donahue, and Percy Liang. Codified audio language modeling learns useful representations for music information retrieval. *arXiv preprint arXiv:2107.05677*, 2021.

[11] Rodrigo Castellon, Chris Donahue, and Percy Liang. Codified audio language modeling learns useful representations for music information retrieval. *arXiv preprint arXiv:2107.05677*, 2021.

[12] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*, 2021.

[13] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[14] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

[15] Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B Dannenberg. Controllable deep melody generation via hierarchical music structure representation. *arXiv preprint arXiv:2109.00663*, 2021.

[16] Shuqi Dai, Zheng Zhang, and Gus G Xia. Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.

[17] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

[18] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *Transactions of the Association for Computational Linguistics*, 2020.

[22] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[23] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, pages 1068–1077. PMLR, 2017.

[24] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6391–6400, 2019.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[26] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[27] Mojtaba Heydari, Frank Cwitkowitz, and Zhiyao Duan. Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking. *arXiv preprint arXiv:2108.03576*, 2021.

[28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[29] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. Mulan: A joint embedding of music audio and natural language. *arXiv preprint arXiv:2208.12415*, 2022.

[30] Qingqing Huang, Aren Jansen, Li Zhang, Daniel PW Ellis, Rif A Saurous, and John Anderson. Large-scale weakly-supervised content embeddings for music recommendation and tagging. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8364–8368. IEEE, 2020.

[31] Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 516–520. IEEE, 2020.

[32] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[33] Corey Kereliuk, Bob L Sturm, and Jan Larsen. Deep learning and music adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, 2015.

[34] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 366–370. IEEE, 2018.

[35] Peter Knees, Ángel Faraldo Pérez, Herrera Boyer, Richard Vogl, Sebastian Böck, Florian Hörschläger, Mickael Le Goff, et al. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR); 2015 Oct 26-30; Málaga, Spain.[Málaga]: International Society for Music Information Retrieval, 2015. p. 364-70.* International Society for Music Information Retrieval (ISMIR), 2015.

[36] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[37] Filip Korzeniowski and Gerhard Widmer. End-to-end musical key estimation using a convolutional neural network. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 966–970. IEEE, 2017.

[38] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pages 387–392, 2009.

[39] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification. *Applied Sciences*, 8(1):150, 2018.

[40] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Yike Guo, and Jie Fu. Mert: Acoustic music understanding model with large-scale self-supervised training, 2023.

[41] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Chenghua Lin, Xingran Chen, Anton Ragni, Hanzhi Yin, Zhijie Hu, Haoyu He, et al. Large-scale pretrained model for self-supervised music audio representation learning, 2022.

[42] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao MA, Chenghua Lin, Xingran Chen, Anton Ragni, Hanzhi Yin, Zhijie Hu, Haoyu He, et al. Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning. In *Ismir 2022 Hybrid Conference*, 2022.

[43] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao MA, Chenghua Lin, Xingran Chen, Anton Ragni, Hanzhi Yin, Zhijie Hu, Haoyu He, et al. Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning. In *ISMIR 2022 Hybrid Conference*, 2022.

[44] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff. Deep contextualized acoustic representations for semi-supervised speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6429–6433. IEEE, 2020.

[45] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE, 2020.

[46] Yinghao Ma, Ruibin Yuan, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, Chenghua Lin, Emmanouil Benetos, Anton Ragni, Norbert Gyenge, et al. On the effectiveness of speech self-supervised learning for music. *arXiv preprint arXiv:2307.05161*, 2023.

[47] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Learning music audio representations via weak language supervision. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 456–460. IEEE, 2022.

[48] Ugo Marchand and Geoffroy Peeters. Swing ratio estimation. In *Digital Audio Effects 2015 (Dafx15)*, 2015.

[49] Matthew C McCallum, Filip Korzeniowski, Sergio Oramas, Fabien Gouyon, and Andreas F Ehmann. Supervised and unsupervised learning of audio representations for music understanding. *arXiv preprint arXiv:2210.03799*, 2022.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[51] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, Fabian-Robert Stöter, Alexandre Défossez, Minseok Kim, Woosung Choi, Chin-Yun Yu, and Kin-Wai Cheuk. Music demixing challenge 2021. *Frontiers in Signal Processing*, 1:18, 2022.

[52] Mateusz Modrzejewski, Piotr Szachewicz, and Przemysław Rokita. Transfer learning with deep neural embeddings for music classification tasks. In *Artificial Intelligence and Soft Computing: 21st International Conference, ICAISC 2022, Zakopane, Poland, June 19–23, 2022, Proceedings, Part I*, pages 72–81. Springer, 2023.

[53] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7354, 2020.

[54] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[55] Santiago Pascual, Mirco Ravanelli, Joan Serra, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *arXiv preprint arXiv:1904.03416*, 2019.

[56] Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654*, 2019.

[57] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, pages 367–372, 2014.

[58] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017.

[59] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 2020.

[60] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[61] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE, 2021.

[62] Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Catharin, Rafael Biazus Mangolin, Valéria Delisandra Feltrim, Marcos Aurélio Domingues, et al. Music4all: A new music database and its applications. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 399–404. IEEE, 2020.

[63] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

[64] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[65] Mohammad Soleymani, Micheal N Caro, Erik M Schmidt, Cheng-Ya Sha, and Yi-Hsuan Yang. 1000 songs for emotional analysis of music. In *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*, pages 1–6, 2013.

[66] Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. *arXiv preprint arXiv:2103.09410*, 2021.

[67] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[70] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

[71] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022.

[72] Luyu Wang, Pauline Luc, Yan Wu, Adria Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, et al. Towards learning universal audio representations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4593–4597. IEEE, 2022.

[73] Ziyu Wang, Yiyi Zhang, Yixiao Zhang, Junyan Jiang, Ruihan Yang, Junbo Zhao, and Gus Xia. Pianotree vae: Structured representation learning for polyphonic music. *arXiv preprint arXiv:2008.07118*, 2020.

[74] Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. Vocalset: A singing voice dataset. In *ISMIR*, pages 468–474, 2018.

[75] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Pablo Bello, and Chao Wang. Multi-task self-supervised pre-training for music classification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 556–560. IEEE, 2021.

[76] Yuya Yamamoto, Juhan Nam, and Hiroko Terasawa. Deformable cnn and imbalance-aware feature learning for singing technique classification. *arXiv preprint arXiv:2206.12230*, 2022.

[77] Dong Yao, Zhou Zhao, Shengyu Zhang, Jieming Zhu, Yudong Zhu, Rui Zhang, and Xiuqiang He. Contrastive learning with positive-negative frame mask for music representation. In *Proceedings of the ACM Web Conference 2022*, pages 2906–2915, 2022.

[78] Ruibin Yuan, Yinghao Ma, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, Le Zhuo, Yiqi Liu, Jiawen Huang, Zeyue Tian, et al. Marble: Music audio representation benchmark for universal evaluation. *arXiv preprint arXiv:2306.10548*, 2023.

[79] Wadhah Zai El Amri, Oliver Tautz, Helge Ritter, and Andrew Melnik. Transfer learning with jukebox for music source separation. In *Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings, Part II*, pages 426–433. Springer, 2022.

[80] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630*, 2021.

[81] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark, 2019.

[82] Yilun Zhao and Jia Guo. Musicoder: A universal music-acoustic encoder based on transformer. In *International Conference on Multimedia Modeling*, pages 417–429. Springer, 2021.

[83] Hongyuan Zhu, Ye Niu, Di Fu, and Hao Wang. Musicbert: A self-supervised learning of music representation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3955–3963, 2021.

[84] Le Zhuo, Ruibin Yuan, Jiahao Pan, Yinghao Ma, Yizhi LI, Ge Zhang, Si Liu, Roger Dannenberg, Jie Fu, Chenghua Lin, et al. Lyricwhiz: Robust multilingual zero-shot lyrics transcription by whispering to chatgpt. *arXiv preprint arXiv:2306.17103*, 2023.

Appendix A

Methodology

Pseudocode for the loss calculation described in the methodology section is presented in Algo. 1.

Algorithm 1 Pseudocode description of the pre-training loss calculation in Python style.

```
def loss_cal(x_batch, x_acoustic_labels):
    # retrieve embeddings for acoustic class
    y_VQ = embedding(x_acoustic_labels)
    # prepare CQT targets
    y_CQT = compute_CQT(x_batch)
    # conduct in-batch mixture
    x_noised = mixture(x_batch)
    # compute the representations
    z = MERT(x_noised)

    # loss calculation
    loss_acoustic = Cross_Entropy(z[mask_idx], y_VQ[mask_idx])
    loss_musical = Mean_Square_Error(z[mask_idx],
        y_CQT[mask_idx])
    return loss_acoustic, loss_musical
```

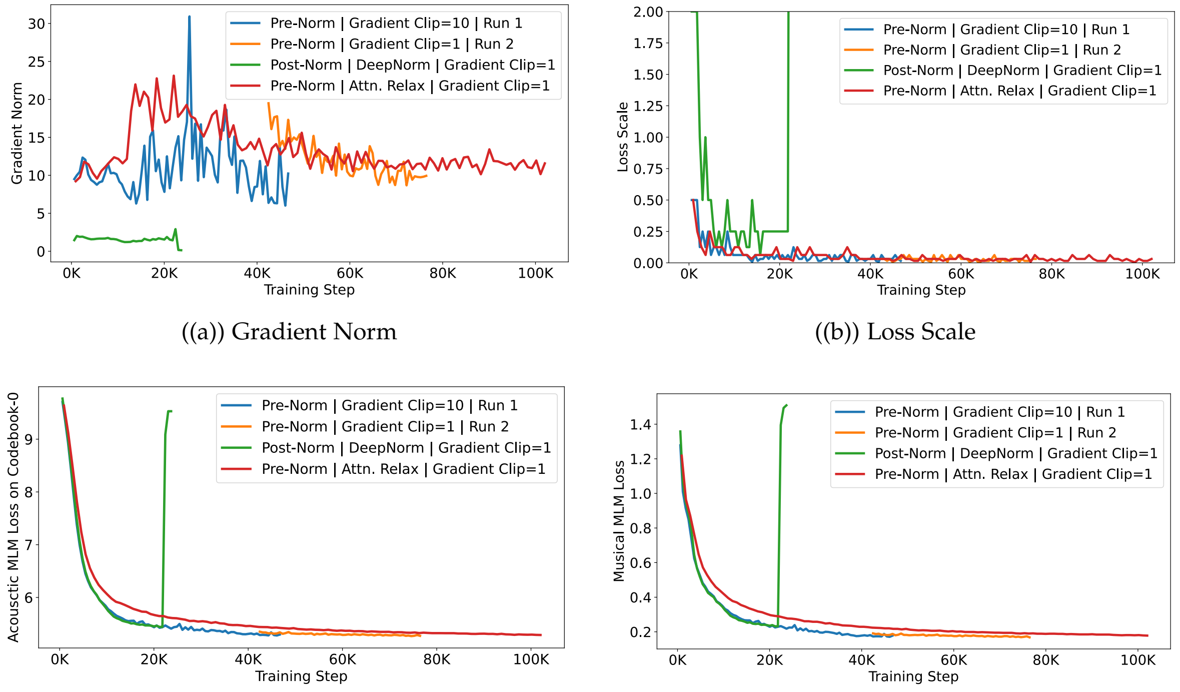
A.0.1 Training Instability

In the experiments of scaling up to MERT-330M under mix precision training (fp16), we have explored several settings and plot the gradient norm, scale of loss, the MLM loss on acoustic targets, and the MLM loss on musical targets (see Fig. A.1).

We first adopt the Pre-LN setting as in the HuBERT [28] x-large model for stable training. However, the training crashed around 50K step under this vanilla solution from the speech model and thus we restart the pre-training at 40K step with gradient clipping threshold reduced from 10.0 to 1.0. The second run of Pre-LN lasted for 40K steps and crashed due to the same reason of reaching minimum loss scale.

We suspect the instability could be brought by the increased depth of the Transformer module. Following the strategies in DeepNorm [71], we tried to alleviate the instability by initializing the Transformer with smaller values and enhancing the residual connection in the Post-LN. Unfortunately, such modification causes model collapse around 20K steps.

We then turned back to the stable Pre-LN setting and leveraged the attention relaxation trick proposed in [12]. The additional scale constant in softmax calculation in the attention module alleviates the overflow problem and allows the final version of MERT-330M model to be trained stably over 100K steps.



((a)) Gradient Norm

((b)) Loss Scale

((c)) Acoustic MLM Loss on Codebook-0

((d)) Music MLM Loss

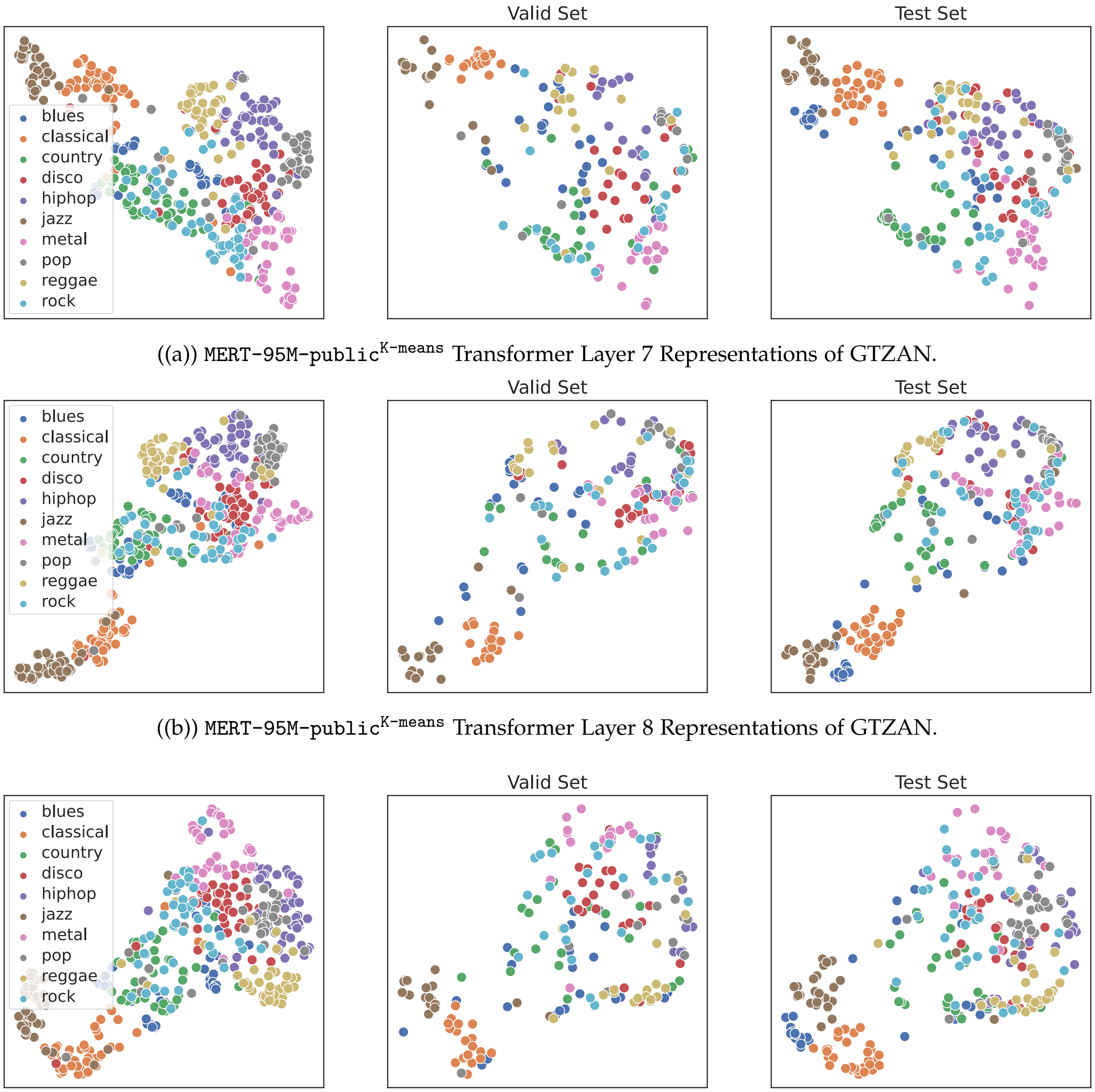
Figure A.1: Illustration of the Training Curves of Trials on Large (330M) Models. Only the acoustic MLM loss on codebook 0 in the RVQ-VAE is shown as the other seven show similar trends.

Appendix B

Representation Visualization

We select two of our checkpoints, MERT-95M-public$^{\texttt{K-means}}$ and MERT-330M$^{\texttt{RVQ-VAE}}$, and visualize the GTZAN representations with genre annotation shown in Fig. B.1, Fig. B.2, Fig. B.3 and Fig. B.4. The top 6 and top 8 transformer output layers are used in the visualization for MERT-95M-public$^{\texttt{K-means}}$ and MERT-330M$^{\texttt{RVQ-VAE}}$, correspondingly. The dimension reduction is achieved by the Uniform Manifold Approximation and Projection (UMAP)[1], whereas the representations from the training set are used to learn the dimension reduction mapping. We observe that representations from both of the checkpoints present a pattern of clustering according to the genre information under different layer settings. Interestingly, the representations from the higher layers do not necessarily show stronger genre-based clustering tendency, which suggests that 1) genre may not be the most abstractive labels for these music examples or 2) the top transformer layers focus more on the MLM pre-training objectives.

[1] https://github.com/lmcinnes/umap

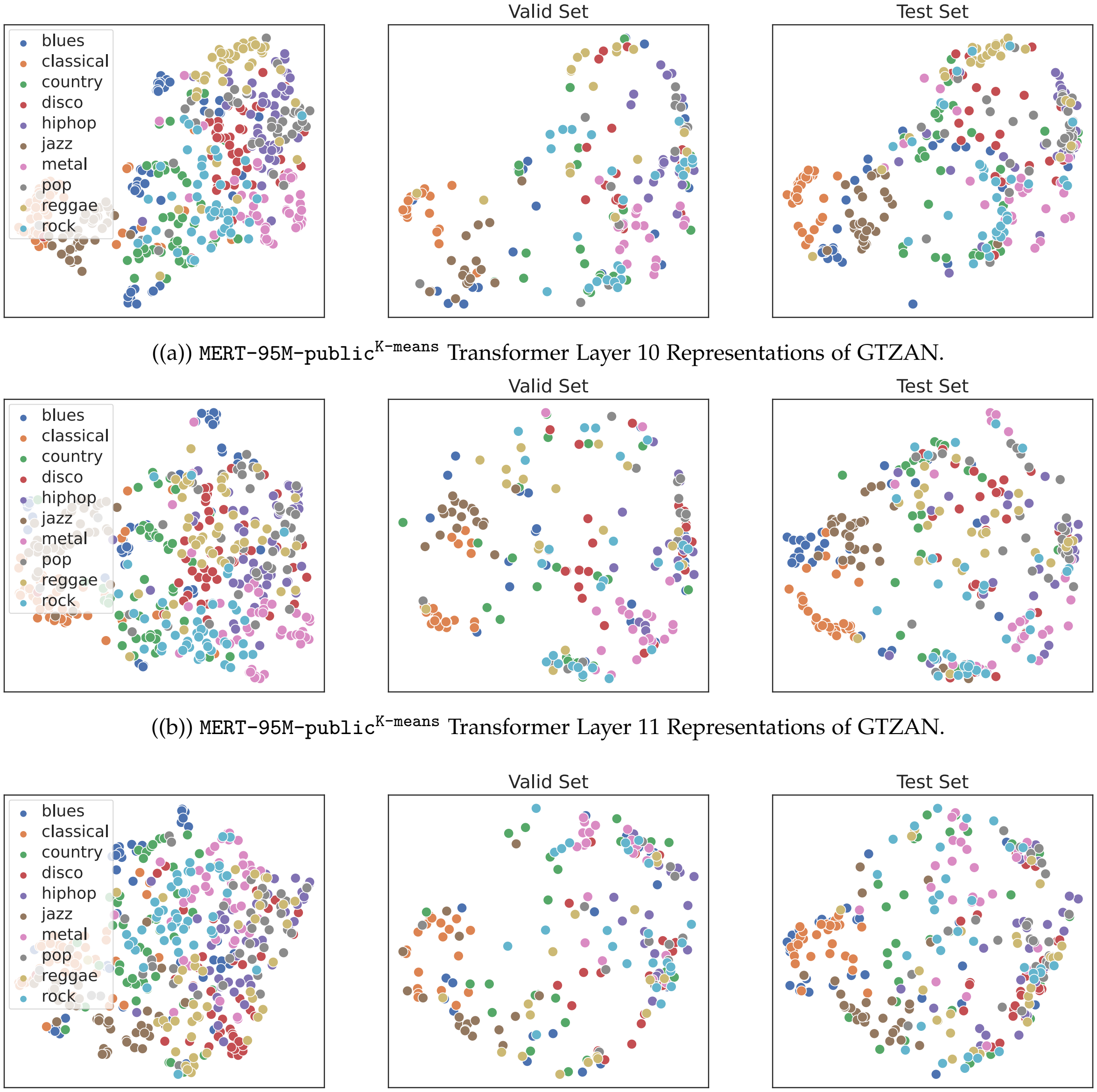


((a)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 7 Representations of GTZAN.

((b)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 8 Representations of GTZAN.

((c)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 9 Representations of GTZAN.

Figure B.1: Illustration of the `MERT-95M-public`$^{\texttt{K-means}}$ Layer 7 to 9 Pre-trained Representations.

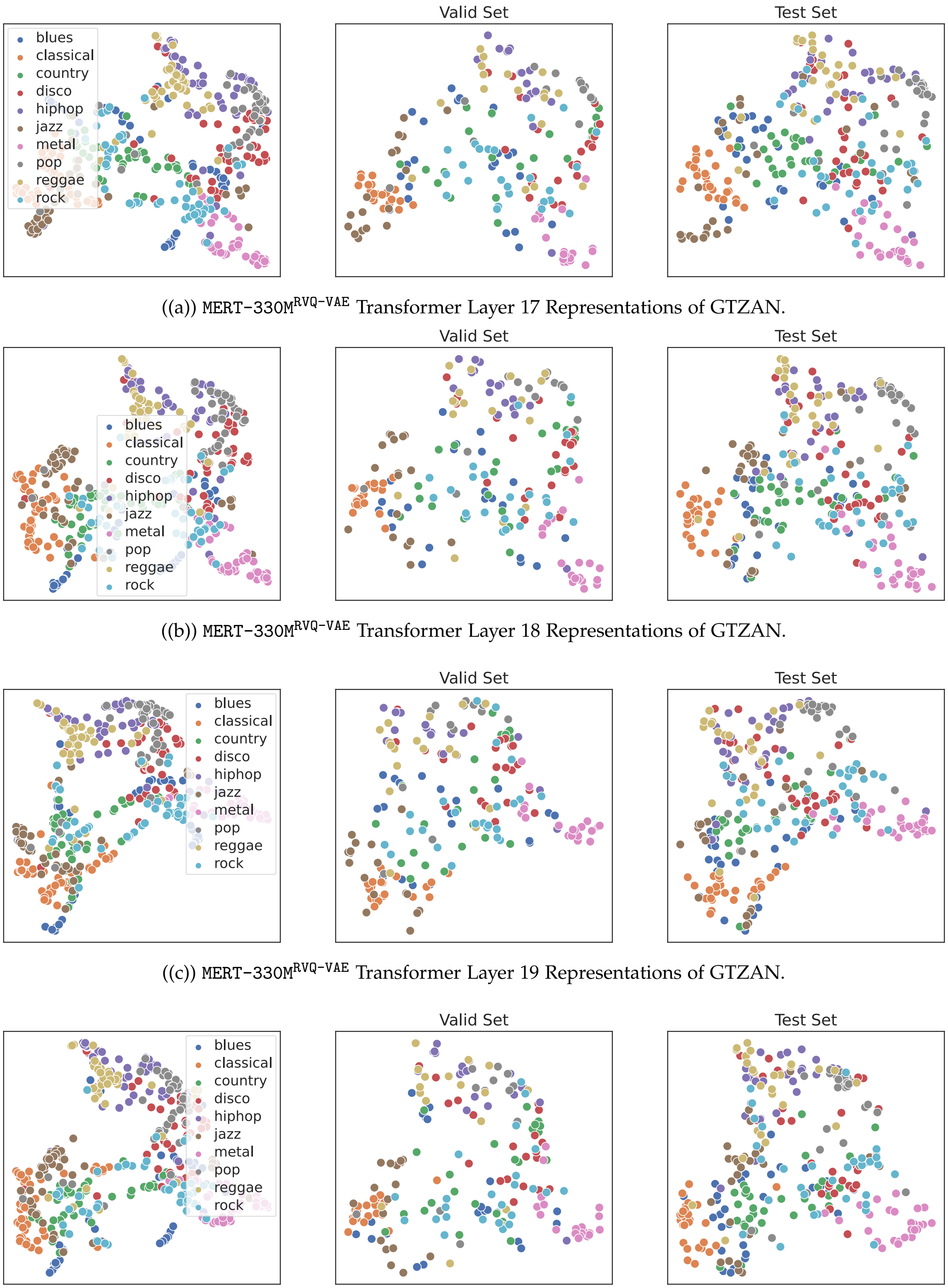


((a)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 10 Representations of GTZAN.

((b)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 11 Representations of GTZAN.

((c)) `MERT-95M-public`$^{\texttt{K-means}}$ Transformer Layer 12 Representations of GTZAN.

Figure B.2: Illustration of the `MERT-95M-public`$^{\texttt{K-means}}$ Layer 10 to 12 Pre-trained Representations.



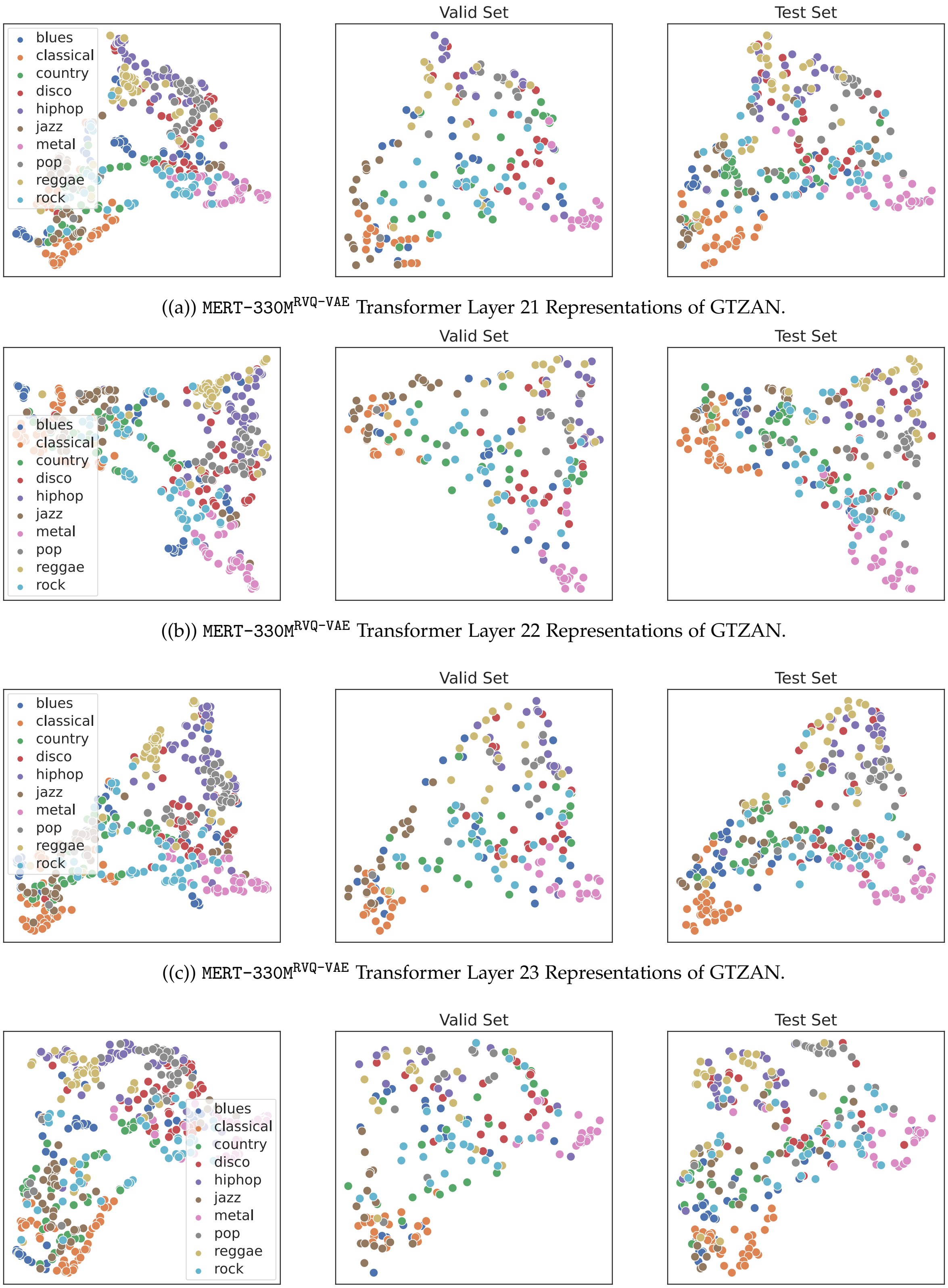
((a)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 17 Representations of GTZAN.

((b)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 18 Representations of GTZAN.

((c)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 19 Representations of GTZAN.

((d)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 20 Representations of GTZAN.

Figure B.3: Illustration of the MERT-330M$^{\text{RVQ-VAE}}$ Layer 17 to 20 Pre-trained Representations.



((a)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 21 Representations of GTZAN.

((b)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 22 Representations of GTZAN.

((c)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 23 Representations of GTZAN.

((d)) MERT-330M$^{\text{RVQ-VAE}}$ Transformer Layer 24 Representations of GTZAN.

Figure B.4: Illustration of the MERT-330M$^{\text{RVQ-VAE}}$ Layer 21 to 24 Pre-trained Representations.