Contrastive Learning with Positive-Negative Frame Mask for Music Representation

Dong Yao
Zhejiang University, China
yaodongai@zju.edu.cn

Zhou Zhao*
Zhejiang University, China
zhaozhou@zju.edu.cn

Shengyu Zhang*
Zhejiang University, China
sy_zhang@zju.edu.cn

Jieming Zhu
Huawei, China
jmzhu@ieee.org

Yudong Zhu
Huawei, China
zhuyudong3@huawei.com

Rui Zhang
www.ruizhang.info, China
rayteam@yeah.net

Xiuqiang He
Huawei, China
hexiuqiang1@huawei.com

ABSTRACT

Self-supervised learning, especially contrastive learning, has made an outstanding contribution to the development of many deep learning research fields. Recently, acoustic signal processing field researchers noticed its success and leveraged contrastive learning for better music representation. Typically, existing approaches maximize the similarity between two distorted audio segments sampled from the same music. In other words, they ensure a semantic agreement at the music level. However, those coarse-grained methods neglect some inessential or noisy elements at the frame level, which may be detrimental to the model to learn the effective representation of music. Towards this end, this paper proposes a novel Positive-nEgative frame mask for Music Representation based on the contrastive learning framework, abbreviated as PEMR. Concretely, PEMR incorporates a Positive-Negative Mask Generation module, which leverages transformer blocks to generate frame masks on Log-Mel Spectrogram. We can generate self-augmented positives and negatives upon the mask by masking important components or inessential components, respectively. We devise a novel contrastive learning objective to accommodate both self-augmented positives/negatives and positives sampled from the same music. We conduct experiments on four public datasets. The experimental results of two music-related downstream tasks, music classification and cover song identification, demonstrate the generalization and transferability of PEMR for music representation learning.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; *Learning latent representations.*

*Corresponding authors.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512011

KEYWORDS

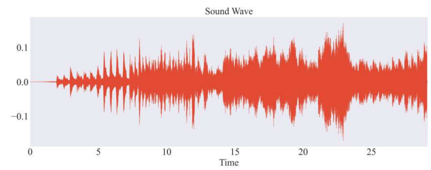
Contrastive Learning, Music Representation, Representation Learning, Attention

ACM Reference Format:
Dong Yao, Zhou Zhao, Shengyu Zhang, Jieming Zhu, Yudong Zhu, Rui Zhang, and Xiuqiang He. 2022. Contrastive Learning with Positive-Negative Frame Mask for Music Representation. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512011

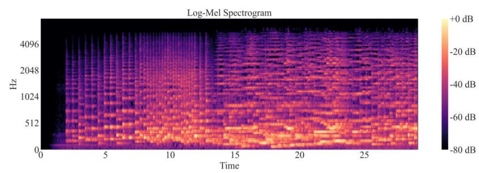
1 INTRODUCTION

Supervised learning has hit a bottleneck. On the one hand, it relies heavily on expensive manual tags and is subject to label errors and false correlations. On the other hand, the amount of labeled data is much smaller than unlabeled data. As a promising alternative, self-supervised learning has drawn massive attention for its data efficiency and generalization ability. Recently, breakthroughs in contrastive learning, such as SimCLR [5], MoCo [16], BYOL [14], Deep Cluster [3], SDCLR [19], shed light on the potential of contrastive learning for learning self-supervised representation. Contrastive learning has increasingly become dominant in self-supervised learning owing to its competitive experimental performance compared with conventional supervised methods.

In Music Information Retrieval (MIR) community, many researchers have made great efforts to learn effective music representation applied in different music-related tasks, such as music classification [7, 8, 24, 27, 32, 34, 35], cover song identification [18, 36–39], chord recognition [6, 22]. However, most of them learn music representation in a supervised manner. Due to the labeled datasets upon which the supervised learning methods depend being costly and time-consuming, the performance of supervised learning methods will be limited. For that reason, some audio research workers have adopted contrastive learning methods to train neural networks. The underlying idea of contrastive learning applied in music is to minimize the distance among the audio segments from the same input while minimizing the similarity among the audio segments from the different inputs. CLMR [30] uses a simple contrastive framework for music representation whose encoder directly encodes raw waveforms of songs. Although it performs well in classification



(a) Raw Waveform



(b) Log-Mel Spectrogram

Figure 1: (a) the raw waveform image of a music track, (b) the corresponding Log-Mel Spectrogram.

downstream tasks, encoding raw waveforms can hardly encode frequency distribution into the final representation of music. To make the model understand the music in time and frequency domain more easily, unlike CLMR, COLA [29] encodes the Log-Mel spectrogram of music so that the time-frequency information can be embedded into music representation. BYOL-A [26], adopting the network structure of BYOL, which owns an online network and a target network, also opt the Log-Mel spectrogram of the music as model input. More details about Log-Mel spectrogram can be find in Section 3.2. Nevertheless, there exists an issue that they encode **all** frames of the spectrogram into music representation space. That would be harmful to the quality of learned music representation, since not all frames impact the music positively. As shown in figure 1(b), the onset of the track may be silent or directly missing, resulting in the absence of valid content of these starting frames. The quality of music downloaded from web is uneven. A song may lose its content at the beginning or any other position, while other songs may contain noisy frames. These frames are unimportant parts for the whole music. On the other hand, we argue that each frame has a different status in characterizing music. For example, the drastic parts of rock music are more appropriate than the mild parts when representing the characteristic of a particular song. In other words, the drastic parts are more representative to the rock music. Therefore, when learning music representation, it is necessary for us to restrict the non-critical parts of music while augmenting the role of crucial parts.

In order to address the above challenge, we propose to mask some frames within a piece of music with a **P**ositive-n**E**gative frame mask for **M**usic **R**epresentation. Specifically, a asymmetrical structure module utilizes the parameters of multi-head attention layers from the transformer encoder to produce the positive-negative mask. The positive mask will erase the existence of inessential frames. Thus, the remaining crucial frames will be encoded and projected into the contrastive learning space to obtain the augmented positive representation. In turn, we can get the counterfactual negative representation by adopting the negative mask. Moreover, we design a contrastive learning objective for positive-negative representation pairs. These masks and loss function can make the model pay more attention to the critical frames and reserve the music's global semantic information while reducing the non-critical frames' adverse effects.

We pre-train the model on several public musical datasets and employ labeled data to train classifiers based on self-supervised representation learned by PEMR. The classifiers achieve the state-of-the-art performance in **music classification** task. To evaluate transferability and generalization capability, we fine-tune the encoder pre-trained in a dataset on another dataset for classification. Besides, we apply the pre-trained encoder in **cover song identification** task and fine-tune it. We can obtain the more advanced performance for cover song identification, through incorporating our pre-trained encoder into the current advanced supervised model. In summary, the contribution of this work is threefold:

- We propose to mask some crucial or inessential elements of music so that the inessential parts will be restricted and the critical parts will be boosted when leaning music representation.
- We devise an asymmetrical mask generation module, generating the positive and negative masks for input music, and a contrastive learning loss function. We incorporate them into the contrastive learning framework for learning more effective music representation.
- The extensive experimental results show that our learned musical representation achieves state-of-the-art performance on a downstream classification task. Furthermore, our learned representation improves the performance of cover song identification, demonstrating its effectiveness and transferability.

2 RELATED WORK

2.1 Contrastive Learning

To solve the problem of the ever-growing unlabeled data that cost a lot of human resources and time, lots of self-supervised methods [3, 4, 10, 11, 13, 20, 28] has been proposed in several areas, typically, Computer Vision, Natural language processing. Since [15] whose approaches contrast positive pairs against negative pairs to learn representation, contrastive learning has attracted a great deal of attention from both academia and the industrial community. Contrastive Predictive Coding [17] is an unsupervised objective that learns predictable representations. CMC [31] is view-agnostic and can scale to any number of views by maximizing mutual information between different views of the same observation for learning representation. MoCo [16] views the contrastive learning as a dictionary look-up to build a dynamic queue including samples of the current mini-batch and the previous mini-batch and a moving-averaged encoder. Another method SimCLR [5], is a simple framework for contrastive learning without a memory bank. Recently, BYOL [14] proposed a new architecture for contrastive learning, which consists of online and target networks. They train the networks only with the various augmented views of an identical image without using negative pairs. Most of the above methods are constructed with twin networks. To avoid collapsed solution and minimize the redundancy, [40] contrast samples from features dimension.

Many researchers in the music community have attempted to apply contrastive learning for learning music representation. [29] designs a common contrastive model for learning general-purpose audio representation. [30] also uses SimCLR [5] framework for pre-training the model. [26] introduces BYOL [14] for audio and achieves advanced results in various downstream tasks.

2.2 Masking Strategy in Music Representation

Masking strategy has played a significant role in the NLP community. The success of BERT [10], which randomly masks some tokens in the input sequence and learns to reconstruct the masked tokens from the output of the transformer encoder, has shown its superiority in learning contextual information among tokens that has attracted the attention of researchers in the audio domain. For example, MusicBERT [41] devised a bar-level masking strategy as the pre-training mechanism to understand symbolic music. Mockingjay [25] is designed to predict the masked frame through jointly conditioning on both past and future contexts. [42] proposes two pre-training objectives, including Contiguous Frames Masking (CFM) and Contiguous Channels Masking (CCM), designed to adapt BERT-like masked reconstruction pre-training to continuous acoustic frame domain.

3 PROPOSED METHOD

The overall architecture of our pre-training framework is shown in Figure 2. Our networks consist of a predicting module which utilizes a transformer encoder to learn contextual correlation among frames, an asymmetrical positive-negative mask generating module, and a contrastive learning module. After pretraining with music datasets, we utilize the **Encoder**, as previous works do [5, 14, 16], to obtain the general music representation for various downstream tasks, such as music classification, cover song identification.

3.1 Sampling and Augmentation.

The function of this unit is to select two segments from the same waveform randomly and apply some augmentation methods in the selected segments. We use the same group of music augmentations as in CLMR [30]. TThe group includes polarity inversion, noise, gain, filter, delay, pitch shift. Each augmentation is randomly selected accordingly to its setting probability.

3.2 Log-Mel Spectrogram

The digital audio signal represents the voltage amplitude of a song varies over time. According to the Fourier theorem, every signal can be decomposed into a set of cosine and sine waves that add up to the original signal, *i.e.*, an audio signal is comprised of several single-frequency sound waves. We use its Mel Spectrogram, generated by Short-Term Fourier Transformation (STFT) and Mel-scale filter banks to capture the time-domain and frequency-domain information of music raw waveform. The Mel-scale aims to mimic the human ear perception function–the human ear's sensitivity varies to different frequencies. In the deep learning domain, [9] trained convolution networks to autonomously discover frequency decompositions from raw audio. For simplicity, we use STFT and Mel-scale filters to obtain the Mel-spectrogram of music and convert it to a logarithmic scale.

3.3 Transformer Encoder

Not all frames of a piece of music play an equivalent role in characterizing songs. Randomly masking the frames of music is a straightforward method to avoid the adverse effect of the inconsequential music frames. However, the critical frames of music may be masked, resulting in the encoder learning the inaccurate representation of the music. Simultaneously, the trivial frames will be retained and encoded, which will deteriorate the learned representation to a certain extent. Therefore, it is indispensable to capture the semantic correlation between the frames and approximately quantify the importance of a single frame to the entire music. Specifically, we use a random masking strategy in the input data before feeding it to the transformer encoder. We then use a predicting layer to recover the masked positions from the output of the transformer encoder, obtaining a predicting loss so that the transformer encoder can learn the correlation of each frame to the input music.

The Transformer encoder uses self-attention mechanisms primarily and learned or sinusoidal position information. Each layer consists of a self-attention sub-layer followed by a position-wise fully connected feed-forward network sub-layer.

We view a frame as a token. In order to make the transformer encoder more stable and accurate when modeling the correlation among all tokens from a music fragment, we train it with a random mask strategy [10, 25, 42]. we denote the frames set of a spectrogram as $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L)$, where $\mathrm{F} \in \mathbb{R}^{L \times D}$. We append a learnable [CLS] token embedding in front of all frames, denoted as $\mathbf{X} = (\mathbf{c}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L)$, where $\mathrm{X} \in \mathbb{R}^{(L+1) \times D}$, so that we can aggregate information of all frames into [CLS] after attention operation. Then, we utilize a multi-head attention mechanism to calculate attention scores between a query and a key and use it for a value, which allows the model to focus on different parts of the frames sequence. Specifically, the formulation of multi-head attention is,

$$\begin{aligned} \mathbf{Y}_{n,h} &= \mathrm{Attention}(\mathbf{Q}_{n,h}, \mathbf{K}_{n,h}, \mathbf{V}_{n,h}) \\ &= \mathrm{Softmax}\left(\frac{\mathbf{Q}_{n,h}\mathbf{K}_{n,h}^T}{\sqrt{D}}\right)\mathbf{V}_{n,h} \end{aligned} \tag{1}$$

where $\mathbf{Q}_{n,h}$, $\mathbf{K}_{n,h}$, $\mathbf{V}_{n,h}$ are the query, key and value respectively. The n and the h are the index of layer and attention head respectively. They are calculated by $\mathbf{Q}_{n,h} = \mathbf{X}\mathbf{W}_{n,h}^Q$, $\mathbf{K}_{n,h} = \mathbf{X}\mathbf{W}_{n,h}^K$ and $\mathbf{V}_{n,h} = \mathbf{X}\mathbf{W}_{n,h}^V$. The $\mathbf{W}_{n,h}^Q$, $\mathbf{W}_{n,h}^K$ and $\mathbf{W}_{n,h}^V \in \mathbb{R}^{D \times D}$, which are the corresponding weight matrices. The attention score between $\mathbf{Q}_{n,h}$ and $\mathbf{K}_{n,h}$ is divided by $\sqrt{D}$ to avoid large values of the dot product. Because the self-attention can not aware of the order, we add a sinusoidal position embedding to the frames sequence before input to self-attention.

The multi-head attention aggregates contextual information through learnable weights, but it is still a linear model. For introducing the non-linearity, multi-head attention output will be fed to the position-wise feed-forward network (FFN) with two layers. Specifically, within the nth layer of transformer encoder, we concatenate the outputs of all attention heads and apply linear transformation to get $\mathbf{Y}_n$, and input it into FFN to obtain output

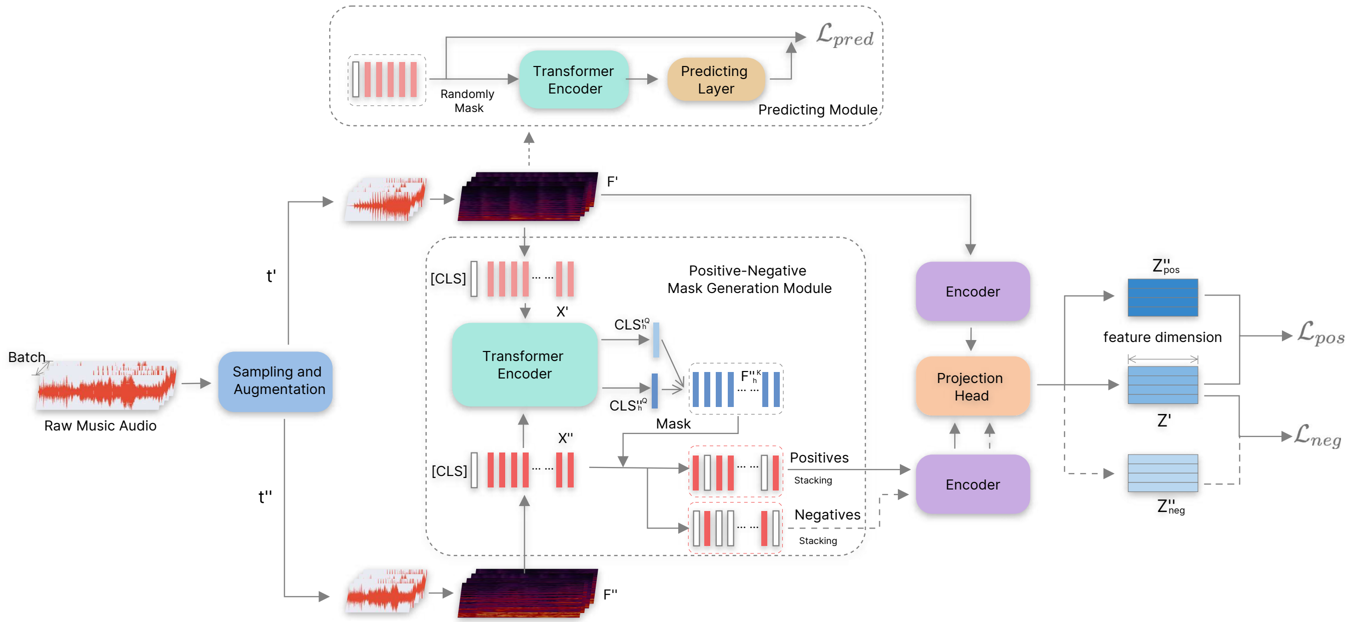


Figure 2: The overall framework of our proposed method for music representation learning. The predicting module is to capture the correlation among frames.

$\mathbf{X}_n$,

$$\mathbf{Y}_n = \text{Concat}(\mathbf{Y}_{n,1}, \mathbf{Y}_{n,2}, \ldots, \mathbf{Y}_{n,h}, \ldots, \mathbf{Y}_{n,H})\mathbf{W}_n \quad (2)$$

$$\mathbf{X}_n = \text{ReLu}(\mathbf{Y}_n\mathbf{W}_{n,1} + \mathbf{b}_{n,1})\mathbf{W}_{n,2} + \mathbf{b}_{n,2} \quad (3)$$

where $\mathbf{W}_n \in \mathbb{R}^{HD\times D}$, $\mathbf{W}_{n,1}$ and $\mathbf{W}_{n,2} \in \mathbb{R}^{D\times D}$, $\mathbf{b}_{n,1}$ and $\mathbf{b}_{n,2} \in \mathbb{R}^{D}$. The H is the number of attention heads. For other details about transformer, such as positional encoding and residual connections, you can find from [33]. We then randomly mask the frames and use FFN to predict the missing content so that we can learn robust contextual information between frames. The protocol of random masking follows [10].

3.4 Generating Positive-Negative Frame Mask

We argue that the crucial frames of a raw waveform can facilitate us learning a more distinct music representation that the network can identify different music more easily and accurately. As we describe above that the noisy parts or inessential parts will be detrimental to the music representation. Reducing the effect of non-critical frames is necessary. Hence, we generate the positive and negative masks to create the augmented positive and counterfactual negative representations. The agreement between positives, the distance between positives and negatives will be optimized by the contrastive learning loss functions. You can see more details in Section 3.6.

We design an asymmetrical module to obtain the positive-negative mask. Firstly, we randomly select two fragments from the same music raw waveform. After applying several augmentation approaches in them, we get their Log-Mel spectrogram produced by stacking a lot of frames. Before inputting them into the transformation encoder, we add a [CLS] token vector in front of their frames sequences. As we illustrate above, the vector $\mathbf{c}$ will contain the information of all frames within a sequence after being encoded by the transformation encoder. Therefore, we use the query vector of $\mathbf{c}'$ and $\mathbf{c}''$ to calculate attention scores between it and the keys of frames $\mathbf{F}''$. The attention scores will be used to select a certain percentage of frames to be masked. Specifically, within the last layer of transformer encoder, we take out the $\mathbf{CLS}_h'^{Q}$ and $\mathbf{CLS}_h''^{Q}$ of added tokens in both branches. Then, we utilize these queries to calculate the attention scores with $\mathbf{F}_h''^{K}$,

$$\mathbf{s} = \frac{1}{2}\sum_{h=1}^{H}(\text{Softmax}(\mathbf{CLS}_h'^{Q} \cdot \frac{\mathbf{F}_h''^{K}}{\sqrt{D}} + \text{Softmax}(\mathbf{CLS}_h''^{Q} \cdot \frac{\mathbf{F}_h''^{K}}{\sqrt{D}})) \quad (4)$$

where $\mathbf{F}_h''^{K}$ is the keys in the hth attention head, $\mathbf{CLS}_h'^{Q}$ and $\mathbf{CLS}_h''^{Q} \in \mathbb{R}^{1\times D}$, $\mathbf{F}_h''^{K} \in \mathbb{R}^{D\times L}$, $\mathbf{s} \in \mathbb{R}^{1\times L}$. The frames with high values in the *Scores* mean that they are crucial to both music fragments under two augmentation views. According to the $\mathbf{s}$, we can screen a certain proportion of the frames with the lower attention weights. The remained crucial frames will reserve the global and local information since the two fragments locate in different positions of the whole music raw waveform. Specifically, we rank the $\mathbf{s}$ in the ascending order and set the value ranked at ratio $\boldsymbol{r}$ as the threshold $\boldsymbol{t}$. The $\boldsymbol{r}$ is the ratio value, and we set it to 10% as the default value. We obtain the positive mask matrix $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_i, \ldots, \mathbf{m}_L)$

as follows,

$$\mathbf{m}_i = \begin{cases} \mathbf{0}, & \mathbf{s}_i < t \\ \mathbf{e}, & others \end{cases} \tag{5}$$

where $\mathbf{e}$ is unit vector, $\mathbf{0}$ is a zero vector. The negative mask $\overline{\mathbf{M}} = 1 - \mathbf{M}$. We add the positive and negative mask to the input frames $\mathbf{F}^{''}$ to obtain the augmented positive frames $\mathbf{F}^{''}_{pos}$ and counterfactual negative frames $\mathbf{F}^{''}_{neg}$ respectively. The $\mathbf{F}^{''}_{pos}$ and $\mathbf{F}^{''}_{neg}$ will be encoded and projected into the positive representation $\mathbf{Z}^{''}_{pos}$ and negative representation $\mathbf{Z}^{''}_{neg}$.

3.5 Encoder and Projection Head

As the familiar setting in contrastive learning [5, 14, 30], we apply a neural network encoder $f(\cdot)$ to extract representation vectors from augmented examples and use a MLP with one hidden as projection head $g(\cdot)$ to map representations to the latent space where contrastive loss is applied. We opt Fully Convolutional Networks (FCN) [7] as our base encoder. The dimensionality of representation vectors from encoder is $D_e = 512$, from projection head is $D_p = 256$.

3.6 Pre-training Objective Function

For training the model, we adopt Huber loss [12], and Barlow Twins loss [40] as our pre-training objective. In section 3.3, a predicting layer is to predict the random disturbed input according to the output of the transformer encoder. The output of the predicting layer is denoted as $\mathbf{P}$. The set $\mathbf{I}$ includes all masked frames' index. We calculate the predicting loss $\mathcal{L}_{pred}$ as follows,

$$\mathcal{L}_{pred} = \sum_{i \in \mathbf{I}} \sum_{j=0}^{D} \text{smooth}_{L_1}(\mathbf{X}_{i,j} - \mathbf{P}_{i,j}), \tag{6}$$

where

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5 \cdot x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{7}$$

L2 loss is more sensitive to outliers due to the square function. To stabilize training, we follow [12] to use L1 loss when |x| is larger than 1. So the $\mathcal{L}_{pred}$ is less sensitive to outliers. In section 3.5, we feed a batch of $\mathbf{F}^{'}$, the augmented positive version and counterfactual negative version of a batch of $\mathbf{F}^{''}$ into encoder and projection head to respectively get $\mathbf{Z}^{'}$, $\mathbf{Z}^{''}_{pos}$, $\mathbf{Z}^{''}_{neg} \in \mathbb{R}^{B \times D_p}$, where B is the value of batch size. The $\mathbf{Z}^{'}$ and $\mathbf{Z}^{''}_{pos}$ are treated as the positive samples in the contrastive space while $\mathbf{Z}^{''}_{neg}$ is the negative samples. We compute the contrastive loss between $\mathbf{Z}^{'}$ and $\mathbf{Z}^{''}_{pos}$, denoted as $\mathcal{L}_{pos}$ in the following manner,

$$\mathcal{L}_{pos} = \sum_{i=0}^{D_p} (1 - \mathbf{U}_{i,i})^2 + \lambda \sum_{i=0}^{D_p} \sum_{j \neq i}^{D_p} \mathbf{U}_{i,j}^2 \tag{8}$$

where $\mathbf{U} \in \mathbb{R}^{D_p \times D_p}$ is the cross-correlation matrix between $\mathbf{Z}^{'}$ and $\mathbf{Z}^{''}_{pos}$. $\mathcal{L}_{pos}$ is the same as BTLoss [40]. Meanwhile, we design a contrastive loss for the negative samples,

$$\mathcal{L}_{neg} = \lambda \sum_{i=0}^{D_p} \mathbf{V}_{i,i}^2 \tag{9}$$

where $\mathbf{V} \in \mathbb{R}^{D_p \times D_p}$ is the cross-correlation matrix between $\mathbf{Z}^{'}$ and $\mathbf{Z}^{''}_{neg}$. The λ is a hyperparameter to trade off the importance of $\mathcal{L}_{neg}$ and the second term of $\mathcal{L}_{pos}$. The loss $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ contrast data samples along the feature dimension which can prevent trivial constant solutions. Our final loss is $\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{pos} + \mathcal{L}_{neg}$.

Table 1: The statistics of all datasets. Music Classification and Cover Song Identification are denoted as MC and CSI, respectively.

Dataset	train	validation	test	task
MagnaTagATune	18,706	1,825	5,329	MC
GTZAN	930	-	-	MC
SHS100K	9,999	-	1,004	CSI
Covers80	-	-	160	CSI

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setting

4.1.1 Dataset. We experiment with several available public datasets ofter used for classification and cover song identification. More details about datasets are illustrated as follows,

Music Classification. MagnaTagATune (MTAT): The annotations of MagnaTagATune were collected by Edith Law's TagATune game [23]. The dataset includes 25863 pieces of music which are 29-seconds-long, and each track has multiple tags. The clips span a broad range of genres like Classical, New Age, Electronica, Rock, Pop, World, Jazz, Blues, Metal, Punk, and more. We split it into train/valid/test with a ratio as [30], and get 18706/1825/5329 tracks; **GTZAN[1] :** The dataset consists of approximately 1000 audio tracks, each 30 seconds long. It contains ten genres.

Cover Song Identification. A cover version is a new performance or recording by a musician other than the original performer of the song. **Second Hand Songs 100K (SHS100K):** We crawled raw audios through youtube-dl[2] using the provided URLs from Github[3]. Due to the copyright, We crawled all songs from YouTube and got 9733 songs. Every song has many cover versions. All cover versions of the 9733 songs add up to 104612. Following the experimental setting of [38], we selected the songs whose number of cover song versions were larger than 5 for training. We randomly selected tracks from the remaining records to construct two subsets for validation and testing, respectively. The ratio among training set, validation set, and testing set is 8:1:1. We get 6000 songs with 84153 versions for training, 1941 songs with their 10456 cover songs for testing; **Covers80[4]:** There are 80 songs exists in Covers80, and every song has 2 cover versions.

4.1.2 Metrics. To evaluate our learned representation for music, we follow the commonly used linear evaluation setting [1, 5, 21, 30], where a linear classifier is trained on an encoder from which parameters are not updated. Moreover, we train a multi-layer perceptron (MLP) to observe if the performance can be better after adding

[1] http://marsyas.info/downloads/datasets.html
[2] https://github.com/ytdl-org/youtube-dl
[3] https://github.com/NovaFrost/SHS100K2
[4] https://labrosa.ee.columbia.edu/projects/coversongs/covers80/

Table 2: The performance of some advanced supervised and self-supervised methods in music classification tasks is all trained on the MagnaTagATune dataset. For the unsupervised models, the scores are obtained by linear classifiers. * represents the performance of an MLP classifier.

	Method	Param	ROC-AUC	PR-AUC
Supervised	1D-CNN	382K	85.6	29.6
	SampleCNN	2394K	88.6	34.4
	Musicnn	228K	89.1	34.9
	Timber CNN	220K	89.3	-
	FCN-4	370K	89.4	-
Self-Supervised	MoCo	370K	87.0	32.1
	MoCo v2	370K	87.9	33.2
	CLMR	2394K	88.5	35.4
	SimCLR	370K	88.7	34.8
	BYOL	370K	89.1	35.8
	PEMR(ours)	370K	**89.6**	**36.9**
Self-Supervised	CLMR*	2394K	89.3	35.9
	MoCo*	370K	89.5	36.3
	MoCo v2*	370K	89.8	36.6
	SimCLR*	370K	89.8	36.9
	BYOL*	370K	89.9	37.0
	PEMR(ours)*	370K	**90.3**	**38.0**

the depth of the classifier. We choose ROC-AUC and PR-AUC to measure the effect of the classifier comprehensively. For the cover song identification task, we use the widely used evaluation metrics [5] mean average precision (MAP), the mean rank of the first correctly identified cover (MR1), and precision at 10 (Precision@10). Precision@10 is the mean ratio of the identical versions recognized successfully in the top 10 ranking list, which is obtained through ranking all records by the similarity between query and references. We calculate scalar products between two music representations to judge their similarity.

4.1.3 Implementation Details. The basic encoders, a full convolution network with 4 layers, share parameters between two branches. The encoder outputs a 512-dimension feature as a representation. An MLP as projection head is used to map the representations to a smaller space. The output dimension of the projection head is 256. It is worth mentioning that the encoders and projection heads of all the unsupervised methods used in the experiments are consistent. We use the Adam optimizer. The learning rate is 0.0003, and weight decay is 1.0×10^{-6}. Others are default. We set the batch size to 64 and train for 300 epochs, taking about 30 hours on a GPU. At the spectrogram extracting stage, the hop size is 128 during time-frequency transformation. STFT is performed using 256-point FFT while the number of mel-bands is set as 128. The transformer encoder consists of 3 layers, and its multi-head attention sub-layer has 3 heads.

[5]https://www.musicir.org/mirex/wiki/2020:Audio_Cover_Song_Identification

4.2 Music Classification

We select some traditional and advanced supervised baselines in music classification and select some state-of-the-art self-supervised baselines in music representation. To make the results more persuasive, we implement several advanced contrastive learning models for music representation.

4.2.1 Linear Evaluation. Table 2 shows the performance comparison in music classification tasks between other approaches, including supervised methods and self-supervised methods, and PEMR. We follow [5, 16] to calculate the number of parameters of encoders in self-supervised methods. We use thop package[6] to obtain the model size. Following a standard linear evaluation setting [5, 14, 16], We use the training set of MTAT to pre-train the unsupervised learning models and train linear classifiers based on the frozen pre-trained encoder for evaluating on the test dataset of MTAT. The applied encoders are the same for all methods except CLMR. CLMR uses sampleCNN [24] to adapt raw waveform. To ensure a faithful comparison, the metric values of baselines are directly copied from their papers, where Timber CNN and FCN-4 do not report PR-AUC values. Our method achieves the best performance under linear evaluation protocol. We attribute the empirical results to the positive-negative frame mask, making the networks preserve the context while concentrating on the critical parts of music. The comparison between CLMR and other self-supervised methods demonstrates the advantage of Log-Mel spectrogram. In addition, we train the MLP to observe whether the performance can be improved when introducing more parameters. The experimental results of PEMR reach the best of 90.3% in ROC-AUC, 37.8% in PR-AUC.

4.2.2 Semi-Supervised Learning. Getting tagged data for deep learning problems often requires skilled human agents. As a result, the costs associated with the labeling process can make a large number of fully labeled training sets infeasible, while obtaining unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Aiming at estimating if our learned music representation can still perform well in the semi-supervised learning classification task, we decrease the percentage of the labeled training data during the fine-tuning stage. Specifically, we randomly sample 1%, 10% labeled data from MTAT training dataset just as [2, 5] do. We directly feed these few labeled data to the pre-trained base encoders and linear classifiers for training. The evaluation results of the previous approaches and PEMR are shown in Table 4. In contrast with other musical self-supervised methods, PEMR can generate more generalizable music representation even if the amount of labeled music for learning is inadequate. Besides, we randomly initialize the base encoder FCN carrying a linear classifier and train it with the same sampled labeled data. Our performance substantially exceeds the model trained from a scratch. The empirical results prove the significance of our pre-trained music representation at a labeled-data-lacked scenario.

4.2.3 Transfer Learning. Specifically, we adopt the whole GTZAN dataset for pre-training the model and employ the MTAT training dataset to train classifiers which will evaluate on the MTAT testing

[6]https://pypi.org/project/thop/

Table 3: We pre-train the models in the GTZAN dataset and transfer their learned parameters to another dataset for training. We evaluate the transfer capacity in both linear evaluation and fine-tuning settings.

Method	Linear Evaluation		Fine-tuned	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
FCN	-	-	89.8	36.8
MoCo	74.8	18.9	89.6	36.6
MoCo v2	78.4	21.3	89.8	36.8
CLMR	81.9	26.2	89.7	36.1
BYOL-A	86.7	32.0	89.3	36.0
COLA	86.8	32.1	89.6	36.6
PEMR(ours)	**87.9**	**33.8**	**90.0**	**37.3**

Table 4: We fine-tune the pre-trained encoders and linear classifiers with different quantities of labeled data.

Method	Label Fraction			
	1%		10%	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
FCN	73.2	19.7	86.3	30.8
MoCo	74.3	18.6	87.4	33.1
MoCo v2	75.3	20.1	87.2	32.4
CLMR	77.3	22.6	87.0	32.9
COLA	73.1	18.6	87.2	32.5
BYOL-A	76.2	22.4	87.6	33.4
PEMR(ours)	**77.3**	**24.4**	**88.0**	**34.0**

Table 5: We experiment with a baseline contrastive framework without the mask. Then, we apply the positive mask and the negative mask in order.

Method	ROC-AUC	PR-AUC
w.o. mask	89.1	36.2
pos. mask	89.4	36.6
pos.+ neg. mask (PEMR)	89.6	36.9

dataset. To reveal the superiority of PEMR more apparently, we compare several self-supervised methods for music representation, including previous approaches and ours. As shown in Figure 3, the effect of the classifiers based on the encoder pre-trained by PEMR is the most advanced under both linear evaluation and fine-tuning settings. More importantly, we can surpass the same network trained in the supervised learning paradigm in the fine-tuning situation. Furthermore, although we freeze the pre-trained encoder when training the classifier, we can also obtain competitive results of 87.9% in ROC-AUC and 33.8% in PR-AUC. The results are comparable to the supervised FCN trained from scratch.

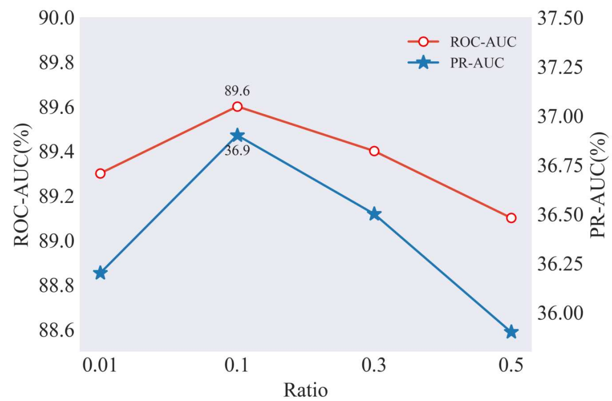


Figure 3: The variation of the performance with different masked ratios.

4.3 Ablation Study

This section will analyze the impact of the augmented positive representation and the counterfactual negative representation generated from the positive-negative mask on learning high-quality music representation. Besides, we modify the vital parameter *ratio* when creating the positive-negative mask to control the proportion of the masked frames in input data. Specifically, the *ratio* ∈ [0.01, 0.1, 0.3, 0.5], we change its value and pre-train the network from scratch for 300 epochs. The experimental results are shown in Table 5 and plotted in Figure 3. We find that,

- We pre-train the baseline model without any masking strategy. Based on the baseline, we only add a positive mask into the model. Their experimental results are shown in Table 5. Generating the augmented positive frames and the counterfactual negative frames is beneficial for the model to learn effective music representation.
- In Figure 3, the model achieves the best result when *ratio* is 0.1 while suffering from performance drops if the value of *ratio* continues growing. We ascribe this phenomenon to: 1) the mild effect of the mask. When *ratio* is too small, for example, *ratio* equals to 0.01, which means little low-scores frames selected to construct the negative frames sequence $\mathbf{F}''_{neg}$. Most of the noisy or inessential frames will be retained in the positive frames sequence $\mathbf{F}''_{pos}$, resulting the model can not focus on the crucial elements. 2) the excessive effect of the mask. The negative frames sequence will contain a large number of the crucial frames if *ratio* is too big. That will destroy the positive frames sequence, causing the model to learn inaccurate music representation.

4.4 Cover Song identification

The task of cover song identification is to identify alternative versions of previous musical works. Since different versions of a song are performed by various artists or musicians and instruments, they may vary significantly in pitch, rhythm, structure, and even in fundamental aspects related to the harmony and melody of the song.

Table 6: Transfer learning for cover song identification. The music representation encoder used in this task is pre-trained with the out-of-domain dataset.

Method	MAP	Precision@10	MR1
	SHS100K-SUB		
Ki-Net	0.112	0.156	68.33
TPP-Net	0.267	0.217	35.75
FCN	0.289	0.230	34.86
CQT-Net	0.446	0.323	**18.09**
Fine-tuned:			
(rand. FCN) + CQT-Net	0.433	0.317	21.13
(pre. FCN) + CQT-Net	**0.484**	**0.341**	20.68
	Covers80		
Ki-Net	0.368	0.052	32.10
TPP-Net	0.5	0.068	17.08
FCN	0.529	0.073	12.50
CQT-Net	0.666	0.077	12.20
Fine-tuned:			
(rand. FCN) + CQT-Net	0.624	0.079	14.43
(pre. FCN) + CQT-Net	**0.668**	**0.081**	**10.52**

Recently, cover song recognition has attracted attention because it has the potential to serve as a benchmark for other musical similarities and retrieval algorithms. Chord analysis, melodic extraction, and musical similarity are all closely related to cover song identification – another area of music analysis where artificial intelligence is used. Before [36], Previous research mostly involved hand-crafted features, which was intolerable when facing large-scale datasets. Given this, [36] proposed deep learning methods, learning to extract features efficiently for cover song identification. [38], and [39] devised TPP-Net and CQT-Net that could be naturally adapted to deal with key transposition in cover songs and designed a training scheme to make their model more robust. We select these advanced models as our baselines of cover song identification task. The main goal of them is to learn the high-quality representation of songs, employing supervised methods. There is still a remaining great developing space for self-supervised learning applying in this task.

The pre-training for music representation will be greatly meaningful if the pre-trained music representation can be transferred to other downstream tasks where training datasets have little labeled data. After pre-training in the MTAT training dataset, we obtain the encoder and fine-tune it with the datasets from the cover song identification domain. The more details are as following: 1) the network we want to train consists of FCN and CQT-Net. 2) the SHS100K training set is provided for the network to fine-tune. 3) we extract music representation through the trained network and evaluate their performance on the SHS100K testing set and Covers80 dataset. It is worth mentioning that **a lot of songs can not be downloaded from YouTube due to the invalid copyright**, resulting in the much difference between our downloaded SHS100K and SHS100K used in previous methods. So we randomly sample the data from SHS100K to construct a subset of SHS100K, namely SHS100K-SUB, and split it into train, validation, test set with the same ratio as [38, 39]. Table 6 exhibits our experimental results. We randomly initialize the parameters of FCN and CQT-Net. After training in the SHS100K-SUB train set, the performance in the SHS100K-SUB test set or Covers80 can not surpass CQT-Net. Nevertheless, we can improve the model's performance on two datasets by incorporating the pre-trained FCN with CQT-Net.

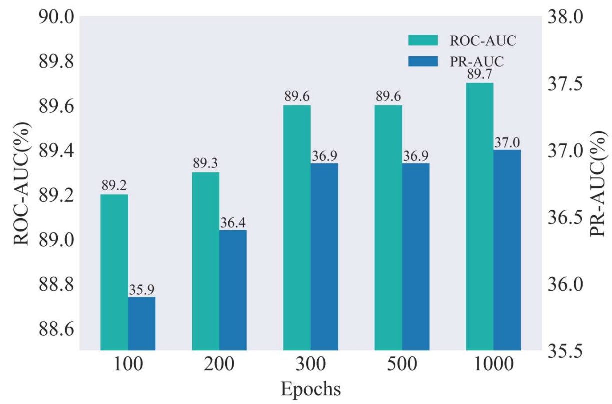


Figure 4: Linear music classifier trained on the top of our pre-trained encoder pre-trained with different epochs.

4.5 Training Epochs

Figure 4 shows the impact of different numbers of training epochs. When training time is relatively short, we find that the training epoch is a critical key influencing the final performance. With more training steps/epochs, the gaps between different epochs decrease or disappear.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose to mask the critical and unimportant or noisy regions of music under the contrastive learning framework so that the model can concentrate on the crucial parts of the music, thus learning the more remarkable and effective representation for music. We devise an asymmetrical module to obtain the positive-negative mask by utilizing the transformation encoder's attention weights. Our pre-trained representation is applied to two music-related downstream tasks, music classification, and cover song identification. The experimental results of two music-related tasks demonstrate the positive-negative mask is beneficial for the model to learn more effective music representation, which has strong generalization ability and transfer ability.

However, there are still many challenge existing in the Music Information Retrieval (MIR) community. Applying pre-trained music representation to music-related areas is a effective way to solve the challenge. We look forward to more related work in the future.

6 ACKNOWLEDGMENTS

The work is supported by the Zhejiang Natural Science Foundation (LR19F020006), and National Natural Science Foundation of China (No. 61836002,62072397,62037001)

REFERENCES

[1] Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems* 32 (2019). arXiv:1906.00910

[2] Lucas Beyer, Xiaohua Zhai, Avital Oliver, and Alexander Kolesnikov. 2019. S4L: Self-supervised semi-supervised learning. *Proceedings of the IEEE International Conference on Computer Vision* 2019-October (2019), 1476–1485. https://doi.org/10.1109/ICCV.2019.00156 arXiv:1905.03670

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11218 LNCS (2018), 139–156. https://doi.org/10.1007/978-3-030-01264-9_9 arXiv:1807.05520

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. NeurIPS (2020), 1–13. arXiv:2006.09882 http://arxiv.org/abs/2006.09882

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv* Figure 1 (2020). arXiv:2002.05709

[6] Tsung Ping Chen and Li Su. 2019. Harmony transformer: Incorporating chord segmentation into harmony recognition. *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019* (2019), 259–267.

[7] Keunwoo Choi, György Fazekas, and Mark Sandler. 2016. Automatic tagging using deep convolutional neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016* (2016), 805–811. arXiv:1606.00298

[8] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2017), 2392–2396. https://doi.org/10.1109/ICASSP.2017.7952585 arXiv:1609.04243

[9] Ieee International Conference and Signal Processing. 2014. END-TO-END LEARNING FOR MUSIC AUDIO Sander Dieleman , Benjamin Schrauwen Electronics and information systems department. *Icassp* (2014), 7014–7018.

[10] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1 (2019), 4171–4186. arXiv:1810.04805

[11] Halvard Fausk and Daniel C. Isaksen. 2007. Improving Language Understanding by Generative Pre-Training. *Homology, Homotopy and Applications* 9, 1 (2007), 399–438. https://doi.org/10.4310/HHA.2007.v9.n1.a16

[12] Ross Girshick. 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169 arXiv:1504.08083

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144. https://doi.org/10.1145/3422622 arXiv:1406.2661

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent: A new approach to self-supervised Learning. 200 (2020). arXiv:2006.07733 http://arxiv.org/abs/2006.07733

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2006), 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975 arXiv:1911.05722

[17] Olivier J. Hénaff, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron Van Den Oord. 2019. Data-efficient image recognition with contrastive predictive coding. *arXiv* 2018 (2019). arXiv:1905.09272

[18] Chaoya Jiang, Deshun Yang, and Xiaoou Chen. 2020. Similarity Learning for Cover Song Identification Using Cross-Similarity Matrices of Multi-Level Deep Sequences. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May (2020), 26–30. https://doi.org/10.1109/ICASSP40776.2020.9053257

[19] Ziyu Jiang, Tianlong Chen, Bobak Mortazavi, and Zhangyang Wang. 2021. Self-Damaging Contrastive Learning. (2021). arXiv:2106.02990 http://arxiv.org/abs/2106.02990

[20] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. 2 (2016), 1–3. arXiv:1611.07308 http://arxiv.org/abs/1611.07308

[21] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (2019), 1920–1929. https://doi.org/10.1109/CVPR.2019.00202 arXiv:1901.09005

[22] Filip Korzeniowski and Gerhard Widmer. 2016. A fully convolutional deep auditory model for musical chord recognition. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP* 2016-November (2016). https://doi.org/10.1109/MLSP.2016.7738895 arXiv:1612.05082

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

[24] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. 2019. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *Proceedings of the 14th Sound and Music Computing Conference 2017, SMC 2017* (2019), 220–226. arXiv:1703.01789

[25] Andy T. Liu, Shu Wen Yang, Po Han Chi, Po Chun Hsu, and Hung Yi Lee. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May (2020), 6419–6423. https://doi.org/10.1109/ICASSP40776.2020.9054458 arXiv:1910.12638

[26] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. 2021. BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation. (2021). arXiv:2103.06695 http://arxiv.org/abs/2103.06695

[27] Jordi Pons and Xavier Serra. 2019. musicnn: Pre-trained convolutional neural networks for music audio tagging. (2019), 4–5. arXiv:1909.06654 http://arxiv.org/abs/1909.06654

[28] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with VQ-VAE-2. *Advances in Neural Information Processing Systems* 32, NeurIPS 2019 (2019), 1–11. arXiv:1906.00446

[29] Aaqib Saeed, David Grangier, and Neil Zeghidour. 2021. Contrastive Learning of General-Purpose Audio Representations. (2021), 3875–3879. https://doi.org/10.1109/icassp39728.2021.9413528 arXiv:2010.10915

[30] Janne Spijkervet and John Ashley Burgoyne. 2021. Contrastive Learning of Musical Representations. (2021). arXiv:2103.09410 http://arxiv.org/abs/2103.09410

[31] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive Multiview Coding. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12356 LNCS (2020), 776–794. https://doi.org/10.1007/978-3-030-58621-8_45 arXiv:1906.05849

[32] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2014. Transfer learning by supervised pre-training for audio-based music classification. *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014* Ismir (2014), 29–34.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 2017-Decem, Nips (2017), 5999–6009. arXiv:1706.03762

[34] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Pablo Bello, and Chao Wang. 2021. Multi-Task Self-Supervised Pre-Training for Music Classification. (2021), 556–560. https://doi.org/10.1109/icassp39728.2021.9414405 arXiv:2102.03229

[35] Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2018. Key-Invariant Convolutional Neural Network Toward Efficient Cover Song Identification. *Proceedings - IEEE International Conference on Multimedia and Expo* 2018-July (2018), 1–6. https://doi.org/10.1109/ICME.2018.8486531

[36] Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2018. Key-Invariant Convolutional Neural Network Toward Efficient Cover Song Identification. *Proceedings - IEEE International Conference on Multimedia and Expo* 2018-July (2018), 1–6. https://doi.org/10.1109/ICME.2018.8486531

[37] Furkan Yesiler, Joan Serra, and Emilia Gomez. 2020. Accurate and Scalable Version Identification Using Musically-Motivated Embeddings. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May, Icassp (2020), 21–25. https://doi.org/10.1109/ICASSP40776.2020.9053793 arXiv:1910.12551

[38] Zhesong Yu, Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2019. Temporal pyramid pooling convolutional neural network for cover song identification. *IJCAI International Joint Conference on Artificial Intelligence* 2019-Augus (2019), 4846–4852. https://doi.org/10.24963/ijcai.2019/673

[39] Zhesong Yu, Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2020. Learning a Representation for Cover Song Identification Using Convolutional Neural Network. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May (2020), 541–545. https://doi.org/10.1109/ICASSP40776.2020.9053839 arXiv:1911.00334

[40] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. (2021). arXiv:2103.03230 http://arxiv.org/abs/2103.03230

[41] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. 2021. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. (2021), 791–800. https://doi.org/10.18653/v1/2021.findings-acl.70 arXiv:2106.05630

[42] Yilun Zhao and Jia Guo. 2021. MusiCoder: A Universal Music-Acoustic Encoder Based on Transformer. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12572 LNCS (2021), 417–429. https://doi.org/10.1007/978-3-030-67832-6_34 arXiv:2008.00781