Learning Music Representations with wav2vec 2.0

Alessandro Ragano
School of Computer Science
University College Dublin
Dublin, Ireland
alessandro.ragano@ucd.ie

Emmanouil Benetos
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
emmanouil.benetos@qmul.ac.uk

Andrew Hines
School of Computer Science
University College Dublin
Dublin, Ireland
andrew.hines@ucd.ie

***Abstract*—Learning music representations that are general-purpose offers the flexibility to finetune several downstream tasks using smaller datasets. The wav2vec 2.0 speech representation model showed promising results in many downstream speech tasks but has been less effective when adapted to music. In this paper, we evaluate whether pre-training wav2vec 2.0 directly on music data can be a better solution instead of finetuning the speech model. We illustrate that when pre-training on music data, the discrete latent representations are able to encode the semantic meaning of musical concepts such as pitch and instrument. Our results show that finetuning wav2vec 2.0 pre-trained on music data allows us to achieve promising results on music classification tasks that are competitive with prior work on audio representations. In addition, the results are superior to the pre-trained model on speech embeddings, demonstrating that wav2vec 2.0 pre-trained on music data can be a promising music representation model.**

***Index Terms*—music representations, self-supervision, pre-training**

I. Introduction

Learning feature representations with deep architectures has shown remarkable success over hand-crafted features in Music Information Retrieval (MIR) [1]. Approaches such as transfer learning from music auto-tagging [2]–[4] allow someone to pre-train neural networks using large datasets and extracting features for downstream MIR tasks such as instrument classification or genre recognition. In this way, downstream MIR tasks can be solved using smaller annotated datasets, which is desired since labeling is costly and difficult to achieve. One issue with auto-tagging models is that they require very large annotated datasets that are still difficult to obtain. To overcome the need for large annotated datasets, new music representation techniques have emerged that do not directly use waveform-related labels emerged. For example, pre-training from language models [5] or using noisy language descriptors of the musical content [6].

A different approach that is based on using proxy tasks to learn representations is self-supervised learning (SSL), where information from input data is extracted to provide labels. This is advantageous since labels can be generated automatically without requiring human intervention. Some SSL models have been proposed for music representations showing competitive performance in several downstream MIR tasks [7]–[13]. Beyond music representation learning, SSL models have grown in popularity for speech representations and downstream tasks such as speaker identification, automatic speech recognition, phoneme recognition, and speech translation [14]. Examples of speech SSL models include wav2vec 2.0 [15] which is a contrastive learning-based approach where the model learns to distinguish a target sample (positive) from distractors (negative). The original model was pre-trained on the LibriSpeech dataset [16] and its success is highlighted by the ability to retain high performance even when dedicated datasets for downstream tasks are very small, e.g., 10 minutes only for speech recognition [15] or 1000 observations for non-intrusive speech quality assessment [17].

The wav2vec 2.0 SSL model has been extensively evaluated for speech tasks. However, its adaptation to music tasks (such as pitch classification or instrument classification) has been limiting. In the NeurIPS challenge HEAR [18], wav2vec 2.0 embeddings are extracted from the model pre-trained on the LibriSpeech dataset and are used as input features without finetuning. Their performance is relatively low in music tasks, even if wav2vec 2.0 speech embeddings can still represent some musical concepts to some degree, such as pitch [18]. Wang et al. [19] have also evaluated wav2vec 2.0 outside of the speech domain. In this case, the authors found that wav2vec 2.0 did not perform well when pre-trained on AudioSet [20], possibly due to the limitation of the masked prediction objective of learning from a dataset more complex than LibriSpeech [19]. More promising results with wav2vec 2.0 outside of the speech domain have been obtained for quality predictions of sound music archives [21] and heart murmur detection [22].

An approach that is still unexplored is pre-training wav2vec 2.0 on music data only for music information retrieval tasks. The transferability of deep networks becomes more challenging when the source and the target tasks have different domains [23] and it has been shown that wav2vec 2.0 might be sensitive to a domain shift. For example, pre-training wav2vec 2.0 with cross-lingual datasets improves performance of ASR systems [24] and finetuning with non-English languages shows

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289_P2 and was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. EB is supported by a Turing Fellowship.

979-8-3503-6021-9/23/$31.00 ©2023 IEEE

a performance drop for speech quality assessment [17].

In this paper, we study whether the domain shift between the pre-trained model and the downstream tasks observed can cause this performance drop in music tasks as reported in the studies above. We explore further the capacity of wav2vec 2.0 features in non-speech tasks by asking the following questions:

1) Does wav2vec 2.0 pre-trained on music encode meaningful music representations, i.e. related to musical concepts such as pitch or instruments?
2) Is it possible to obtain competitive performance on MIR tasks when finetuning wav2vec 2.0 pre-trained on music?
3) Can we establish if wav2vec 2.0 is a potential candidate model for music tasks other than speech?

The paper is structured as follows. In Section 2 we illustrate how we pre-train wav2vec 2.0 with music data. Section 3 is dedicated to the analysis of the features learned by wav2vec 2.0. We show whether the information encoded in the codebooks is related to music labels and we compare the encoded representations in the continuous layers of wav2vec 2.0 pre-trained on music with the information encoded in the original speech model. Section 4 shows the results of finetuning wav2vec 2.0 pre-trained on music on three MIR tasks: instrument classification, pitch classification[1] and singing pitch classification. The singing domain is evaluated since transferring wav2vec 2.0 embeddings from the speech domain to the singing domain has been proposed for automatic lyric transcription [25] but it is still unexplored in other singing analysis tasks such as pitch classification. In Section 5 we discuss whether wav2vec 2.0 pre-trained on music provides promising potential for broader downstream MIR tasks.

II. Method

A. Pre-Trained Model

The wav2vec 2.0 model can be summarized in the following blocks:

1) A *feature encoder* $f : \mathbb{X} \mapsto \mathbb{Z}$ that converts input audio chunks of 20 milliseconds X into a sequence of latent speech representations $\mathbb{Z} = \{z_1, z_2, ..., z_T\}$ for T timesteps. The encoder consists of 7 1D convolutional layers, each with 512 filters.
2) A *context network* $g : \mathbb{Z} \mapsto \mathbb{C}$ based on the Transformer architecture [26] that builds context representations for each audio segment that capture the entire audio sequence $\mathbb{C} = \{c_1, c_2, ..., c_T\}$
3) A *quantization module* that transforms encoder output representations into discrete speech representations $\mathbb{C} \mapsto \mathbb{Q}$. The discrete latent features are learned with product quantization and are needed to create targets for the loss function, but they are not used as input for the context network. A vector that concatenates an entry from each of the 2 codebooks is linearly transformed to get the quantized representations $\mathbb{Q} = \{q_1, q_2, ..., q_T\}$. The Gumbel-Softmax is used to choose the codebook entries in a differentiable way.
4) A *contrastive loss function* is used to learn how to identify the true quantized speech representation from 100 quantized negative samples that are uniformly sampled. Given an audio chunk at time step t, the model compares the cosine similarity between the Transformer output at time step t and the quantized speech representation in the same step t against the similarity with negative distractors. The high similarity with the negative samples is penalized by contrastive loss. The latent speech representation at the time step t created by the feature encoder is masked before being fed to the Transformer-based context network. Negative samples are sampled from other masked time steps of the same utterance.

We use the BASE model configuration [15], which consists of 12 Transformer blocks and produces 768-dimensional feature vectors.

To learn music representations, we pre-trained wav2vec 2.0 on the MusicNet dataset [27]. MusicNet consists of $\approx$ 34 hours of audio across 330 classical music recordings provided as raw waveforms, covering 11 musical instruments. To pre-train wav2vec 2.0 we use the fairseq toolkit [28]. The data is split into overlapped segments of 20 seconds, whose length is recommended in the fairseq repository instructions. To increase the dataset size we take overlapped segments with a hop size equal to 10 seconds collecting $\approx$ 65 hours of audio in total. The dataset that we used to pre-train represents only $\approx 7\%$ of the LibriSpeech dataset size which is the one used to pre-train wav2vec 2.0 for speech [15]. However, we will show that this is sufficient to address whether wav2vec 2.0 learns meaningful music representations. The model was trained for 1790 epochs and it took 7 days on the NVIDIA A100 64GB GPU.

B. Finetuning

Evaluation of downstream tasks is performed on the NSynth dataset [29] using the original train, validation, and test splits. The NSynth dataset includes 305,979 samples of 4 seconds. Two tasks are evaluated on this dataset, pitch classification 1 and instrument classification. Pitch labels on the isolated note recordings are provided as MIDI numbers. Instrument labels represent the instrument family and include the following 11 instruments: bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth_lead, vocal. Notice that synth_lead is only present in the training set, which makes validation and test splits made of 10 instrument classes.

The output of the last Transformer block is a matrix of size $(n \times l)$ where n is the number of time frames and l is the size of the feature vector equal to 768. To remove the time dimension, we simply average across time, obtaining an l-dimensional vector at the output. The latter is connected to a linear layer that consists of the number of output neurons equal to the number of classes of the task: 112 neurons for pitch classification and 11 neurons for instrument classification.

[1] In this paper, we use the term pitch classification since the NSynth dataset is made of isolated note segments. This is different from the more common term "pitch detection" where note segments are not isolated.

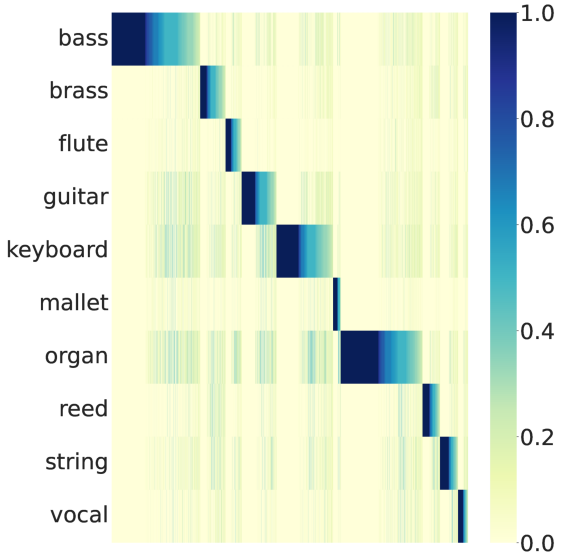

Fig. 1. Co-occurrence between the discrete latent representations and instrument family labels on the NSynth test set.

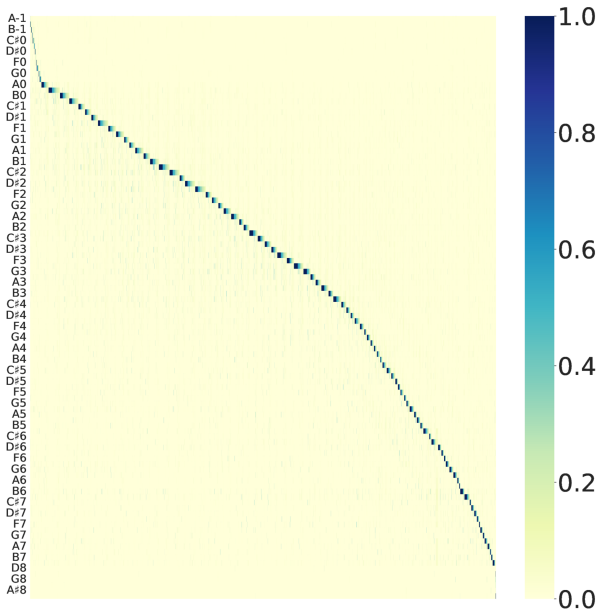

Fig. 2. Co-occurrence between the discrete latent representations and pitch classes on the NSynth test set.

The pre-trained wav2vec 2.0 model with music data is used in 3 different configurations for the downstream tasks: 1) finetuning (FT1) the entire network, 2) finetuning (FT2) the context network (Transformer) while keeping the feature encoder frozen, 3) Freezing both feature and context networks and doing a simple feature extraction (FE) which consists of training only the output linear layer. Finetuning on models FT1 and FT2 is performed using the Adam optimizer with a learning rate of 0.00001 for the pre-trained part and 0.0001 for the output layer. The FE model is trained using the Adam optimizer with a learning rate of 0.001 and only the weights of the output linear layer are optimized. In all 3 configurations, training is stopped if the average loss in the validation set does not decrease for 10 epochs. The cross-entropy loss is used for classification.

III. Feature Analysis

Our first research question in Section 1 asked whether wav2vec 2.0 learns meaningful representations when pre-trained on music data. We first explored whether the learned discrete latent representations used in the loss function encode a semantic meaning related to musical concepts. The discrete representations are an important step in wav2vec 2.0 since learning a finite set of discrete audio units encourages the model not to learn all the variations in the data when minimizing the contrastive loss. We use the NSynth dataset to compute the co-occurrence between both pitch and instrument family labels and the discrete latent features produced by wav2vec 2.0 pre-trained on MusicNet without finetuning.

Figure 1 and Figure 2 show that discrete latent representations specialize in both instrument and pitch classes, respectively. Many latents co-occur with bass, which is the most frequent class in the NSynth test. The discrete latent representations share a similar pattern as the wav2vec 2.0 speech model, where the encoded semantic meaning of the codebooks has been shown to be represented by phonemes [15].

A deeper insight in the analysis of the wav2vec 2.0 features can be obtained by analyzing the Transformer layers. Given a masked latent representation z_t, the objective of the model is to learn a context representation c_t in order to correctly guess the quantized representation q_t among the negative samples. For this reason, it should be expected that the final layers of the Transformer should have higher similarity with the Transformer input. This behaviour should be observed regardless of the input signal type (speech or music). To confirm whether the Transformer layers evolve in the pre-trained model as expected, we follow the same approach of Pasad et al. [30] where they observed this phenomenon occurring in the wav2vec 2.0 pre-trained on speech. We computed the canonical correlation analysis (CCA) between each Transformer layer and the output of the feature encoder. Given a matrix $W \in \mathbb{R}^{n \times k}$ and $Y \in \mathbb{R}^{n \times j}$ with $k < j$, CCA finds two basis such that when the matrices are projected onto the basis their correlation is the highest. More specifically, the CCA is calculated as follows:

$$\rho_i = \max_{u_w^i, u_y^i} corr(W u_w^i, Y u_y^i), \tag{1}$$

$$CCA(W, Y) = \frac{\sum_{i=1}^{k} \rho_i}{k} \tag{2}$$

where ρ_i represents the i-th canonical correlation coefficient, u_w^i and u_y^i are the vectors found by CCA that maximize

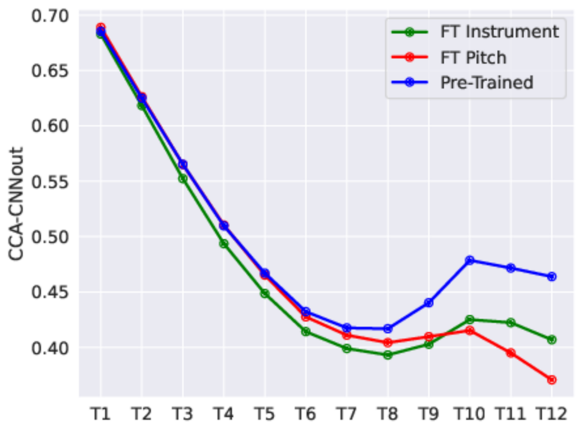

Fig. 3. Evolution of the Transformer layers of the pre-trained model and the finetuned models using PWCCA between each layer and the output of the feature encoder (CNN).

the canonical weights, and the final CCA is obtained with the average. In our analysis, the matrices are represented by the feature vectors at each timestep. We use a variant projection weighted canonical correlation analysis (PWCCA) [31] that is less sensitive to perturbation since it uses a weighted mean to assign a higher weight to the correlation coefficients that have more importance.

The PWCCA is calculated using the FT1 approach where the feature encoder is frozen, and by using frames extracted from the MusicNet dataset. Due to the high computational effort, we take 4 seconds in the middle of each MusicNet observation using half of the dataset size. Figure 3 shows that the pre-trained model attempts to reconstruct the input features (i.e. the output of the feature encoder) while the similarity between the final layers of the finetuned models tend to be lower than the pre-trained model. This confirms that the evolution of the Transformer layers with respect to the feature encoder output is the same in both the speech model (Pasad et al. [30]) and the music model (this study), which is aligned with the objective of wav2vec 2.0.

IV. Downstream Tasks

The second research question in Section 1 asked whether finetuning wav2vec 2.0 pre-trained on music data shows competitive performance in downstream MIR tasks. The performance of wav2vec 2.0 pre-trained on music data is evaluated on pitch and instrument classification using the NSynth test set and singing pitch classification using VocalSet [32]. The model is compared with previous works as shown in Table I.

For instrument pitch classification, we use CREPE [33] which is the best pitch classifier in the HEAR challenge [18], SF NFNet-50 which is the best model in a comparison of audio representations reported by Wang et al. [19], and features extracted from wav2vec 2.0 pre-trained on LibriSpeech which is fundamental to understanding the differences with wav2vec 2.0 pre-trained on music. All the models, except for CREPE, are designed to learn general-purpose audio representations.

For instrument classification, we consider MuLaP [6] which learns music representations by using weak supervision from noisy language descriptors of the musical content, the work of Favory et al. [34] that we call contextual tag embeddings (CTE) where the learned audio representations are aligned to music tags, SF NFNet-50 which is also the best model for instrument classification in the same study from Wang et al. [19], and the feature extracted from wav2vec 2.0 pre-trained on LibriSpeech.

It should be noted that the models reported in Table I have some differences that do not allow for direct comparisons such as pre-training datasets, supervision strategies, hyper-parameters, and strategies to use the learned features. However, the choice of the above prior work models helps us to contextualize the results obtained with wav2vec 2.0 pre-trained on music.

The results in Table I show that pre-training wav2vec 2.0 on music shows comparable results with prior work and improvement over wav2vec 2.0 pre-trained on speech. For pitch classification, finetuning the entire wav2vec 2.0 pre-trained on music achieves the best results together with CREPE and it is the best model among the ones trained to learn general-purpose audio representations. Also, extracting features from the music model shows an 11% increase over the speech embeddings, indicating that pre-training wav2vec 2.0 on music is the contributing factor to the observed performance improvement. For instrument classification, wav2vec 2.0 pre-trained on music is the second-best model after SF NFNet-50 and it shows better results than MuLaP and CTE. We also observe that feature extraction of wav2vec 2.0 from the music model has significant improvement over the speech model, which confirms the positive contribution of the music data used in the pre-training phase. An important aspect to consider is that we pre-trained wav2vec 2.0 on a relatively smaller dataset which still shows competitive results with the other approaches that are pre-trained on larger datasets.

A. Singing Pitch Classification

For singing pitch classification we use VocalSet [32] which consists of monophonic tracks of arpeggios, scales, long tones, and melodic excerpts recorded by nine female and eleven male professional singers. The dataset includes more than 10 hours of recordings, seventeen singing techniques, such as fast, articulated forte, and speaking in rhythm and three music excerpts 1) Row, Row, Row Your Boat, 2) Caro Mio Ben, and 3) Dona Nobis Pacem. To evaluate pitch classification on VocalSet we use the pitch labels created by Faghih and Timoney [35]. They estimated pitch with pYIN [36] and then manually corrected pYIN mistakes using 3 annotators. Estimated pitch is reported with average F0, average standard deviation F0, and median F0 and no significant pitch value differences were found [35]. Therefore, we use the average F0 as the ground truth. In addition, a total of 24.5% of the recordings have been discarded due to singer mistakes [35].

Pitch detection consists of tracking a time-varying pitch. Similarly to what we have done for the NSynth test set,

TABLE I
PERFORMANCE EVALUATION USING RAW PITCH ACCURACY (RP) ON VOCALSET AND CLASS ACCURACY ON NSYNTH TEST SET. THE RESULTS OF THE BASELINE MODEL ARE TAKEN FROM [18]$^{+}$, [19]$^{++}$ [6]$^{+++}$

	VocalSet		**NSynth Test**		**Pre-Training Data**	
	RP¼	RP½	Pitch	Instr.	Type	≈Hours
w2v Music FE	39.1	49.7	76.0	49.0	Music	65
w2v Music FT1	44.5	64.2	82.0	70.0	Music	65
w2v Music FT2	67.1	86.2	90.0	75.0	Music	65
CREPE [33]	92.8	96.1	90.0^{+}			
SF NFNet-50 [19]			88.0^{++}	78.2^{++}	Audio	5800
w2v Speech FE [15]			65.0^{+}		Speech	960
w2v Speech FE [15]			35.0^{++}	40.2^{++}	Speech	960
MuLaP [6]				71.7^{+++}		
CTE [34]				70.0^{+++}	Music	562

we perform pitch classification instead of pitch detection. Unlike NSynth, we do not have isolated note recordings so we perform a segmentation step first by using the onset and offset annotations of VocalSet provided by [35] to extract frames where pitch variations are small. In addition, we consider only frames labelled with 'Sound' [35] i.e. frames where pitch is present. This means that we ignore pitch transition frames and unpitched frames. Onset and offsets locations are estimated in 4 different ways but no statistically significant difference was found [35] so we take the first approach called 'extended 1'. Finally, we found that male singer 9 annotations were also located beyond the duration of the audio tracks, so we discarded tracks performed by this singer.

Singing pitch classification performance is evaluated using raw pitch accuracy which measures the number of correct predicted samples over the total number of samples within a pitch error tolerance. Specifically, we use RP¼ where correct pitch estimates are the ones within a quarter tone from the ground truth and RP½ where correct predictions are the ones within a semitone from the ground truth. The output of the proposed models is a class corresponding to the MIDI code which is converted into Hz values to calculate performance. As a baseline, we use CREPE [33] which is the best pitch classifier in the HEAR challenge [18]. We take the average of all the estimated pitches within the stationary segment extracted with the onset and offset locations as described above. The results in Table I suggest the same trend of the tasks evaluated on the NSynth set above i.e., finetuning the whole network is better than feature extraction or partial finetuning. The relevant gap between RP½ and RP¼ indicates that wav2vec models often classify pitches that are a semitone away from the ground truth. This means that mispredictions are not random and that might be specifically related to the singing domain. The fact that we pre-trained by only using instrumental classical music might indicate why singing pitch classification might be less robust than instrument pitch classification.

V. DISCUSSION & CONCLUSIONS

In this paper, our aim was to study the potential of wav2vec 2.0 for learning meaningful representations from music data. We pre-trained wav2vec 2.0 on music data and evaluated the model on pitch and instrument classification. We demonstrated that wav2vec 2.0 encodes semantic meaning related to musical concepts in the discrete latent representations and that the Transformer layer behaviour is the same of the speech model. We showed that finetuning wav2vec 2.0 pre-trained on music has significant improvement over the original model pre-trained on speech and other audio-representations models. We posed the question: is wav2vec 2.0 pre-trained on music a potential model for learning general-purpose music representations? Our results and analysis support further application of wav2vec 2.0 with music pre-training for broader downstream MIR tasks.

Specifically, we propose to extend these findings by performing a direct comparison with the other models and addressing the following: (i) we pre-trained the model using a small dataset which was sufficient for the questions addressed in this paper but not for general-purpose audio representations that require pre-training with much larger datasets; (ii) the evaluation of the MIR task was conducted using monophonic datasets (NSynth, VocalSet) so the generalization for downstream polyphonic tasks should be explored; (iii) a broader mix of genres in the pre-trained dataset should be explored as MusicNet includes Western classical music and also non-Western music; (iv) the model hyperparameters were not adjusted or optimized and may be better suited to speech than to music; (iv) the model has been trained for a fixed number of epochs due to GPU capacity constraints but more training epochs can be used e.g. by monitoring the contrastive loss with a validation set.

REFERENCES

[1] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: New directions for music informatics," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 461–481, 2013.

[2] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*. International Society for Music Information Retrieval, 2017, pp. 141–149.

[3] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proc. of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014*, 2014.

[4] J. Pons and X. Serra, "musicnn: pre-trained convolutional neural networks for music audio tagging," in *Late-breaking/demo session in 20th International Society for Music Information Retrieval Conference (LBD-ISMIR2019)*, 2019.

[5] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *Proc. of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, 2021.

[6] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Learning music audio representations via weak language supervision," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 456–460.

[7] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, "Multi-task self-supervised pre-training for music classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 556–560.

[8] H. Zhu, Y. Niu, D. Fu, and H. Wang, "Musicbert: A self-supervised learning of music representation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3955–3963.

[9] A. N. Carr, Q. Berthet, M. Blondel, O. Teboul, and N. Zeghidour, "Self-supervised learning of audio representations from permutations with differentiable ranking," *IEEE Signal Processing Letters*, vol. 28, pp. 708–712, 2021.

[10] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *Proc. of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, 2021.

[11] Y. Li, R. Yuan, G. Zhang, Y. MA, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, "Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning," in *ISMIR 2022 Hybrid Conference*, 2022.

[12] Y. Ma, R. Yuan, Y. Li, G. Zhang, X. Chen, H. Yin, C. Lin, E. Benetos, A. Ragni, N. Gyenge *et al.*, "On the effectiveness of speech self-supervised learning for music," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[13] L. Pepino, P. Riera, and L. Ferrer, "Encodecmae: Leveraging neural codecs for universal audio representation learning," *arXiv preprint arXiv:2309.07391*, 2023.

[14] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe *et al.*, "Self-supervised speech representation learning: A review," *arXiv preprint arXiv:2205.10643*, 2022.

[15] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[17] H. Becerra, A. Ragano, and A. Hines, "Exploring the influence of fine-tuning data on wav2vec 2.0 model for blind speech quality prediction," *Proc. Interspeech*, pp. 4088–4092, 2022.

[18] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally *et al.*, "Hear: Holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022, pp. 125–145.

[19] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J.-B. Alayrac, S. Dieleman, J. Carreira *et al.*, "Towards learning universal audio representations," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4593–4597.

[20] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[21] A. Ragano, E. Benetos, and A. Hines, "Audio quality assessment of vinyl music collections using self-supervised learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[22] D. S. Panah, A. Hines, and S. McKeever, "Exploring wav2vec 2.0 model for heart murmur detection," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 1010–1014.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[24] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *Ninth International Conference on Learning Representations (ICLR)*, 2021.

[25] L. Ou, X. Gu, and Y. Wang, "Towards transfer learning of wav2vec 2.0 for automatic lyric transcription," *Proc. of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022*, 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[28] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 48–53.

[29] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[30] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.

[31] A. Morcos, M. Raghu, and S. Bengio, "Insights on representational similarity in neural networks with canonical correlation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[32] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset." in *Proc. of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, 2018, pp. 468–474.

[33] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[34] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "Learning contextual tag embeddings for cross-modal alignment of audio and tags," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 596–600.

[35] B. Faghih and J. Timoney, "Annotated-vocalset: A singing voice dataset," *Applied Sciences*, vol. 12, no. 18, p. 9257, 2022.

[36] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014, pp. 659–663.