MULTI-TASK SELF-SUPERVISED PRE-TRAINING FOR MUSIC CLASSIFICATION

Ho-Hsiang Wu[1,⋆], *Chieh-Chi Kao*[2], *Qingming Tang*[2], *Ming Sun*[2]
Brian McFee[1], *Juan Pablo Bello*[1], *Chao Wang*[2]

[1] Music and Audio Research Laboratory, New York University, USA
[2] Alexa Speech, Amazon

ABSTRACT

Deep learning is very data hungry, and supervised learning especially requires massive labeled data to work well. Machine listening research often suffers from limited labeled data problem, as human annotations are costly to acquire, and annotations for audio are time consuming and less intuitive. Besides, models learned from labeled dataset often embed biases specific to that particular dataset. Therefore, unsupervised learning techniques become popular approaches in solving machine listening problems. Particularly, a self-supervised learning technique utilizing reconstructions of multiple hand-crafted audio features has shown promising results when it is applied to speech domain such as emotion recognition and automatic speech recognition (ASR). In this paper, we apply self-supervised and multi-task learning methods for pre-training music encoders, and explore various design choices including encoder architectures, weighting mechanisms to combine losses from multiple tasks, and worker selections of pretext tasks. We investigate how these design choices interact with various downstream music classification tasks. We find that using various music specific workers altogether with weighting mechanisms to balance the losses during pre-training helps improve and generalize to the downstream tasks.

***Index Terms*—** Self-supervised learning, multi-task learning, music classification

1. INTRODUCTION

Deep learning has shown great successes with end-to-end learned representations replacing hand-crafted features in various machine perception fields, including computer vision, natural language processing and machine listening, especially in supervised learning paradigm. However, unlike ImageNet for computer vision, which contains millions of labeled images, human annotated datasets for machine listening are usually small [1]. Therefore, learning from limited labeled data [2] is especially important. There are existing methods such as transfer learning [3] and domain adaptation, where models learned from different tasks with larger datasets are transferred and fine-tuned to another task/domain, and unsupervised learning [4, 5, 6], such as generative models [7, 8], where data distribution is often learned through reconstruction of the signal.

Self-supervised learning [9, 10, 11, 12], as one sub-field of unsupervised learning, exploits the structure of the input data to provide supervision signals. It has become more popular in recent years, showing good improvement in multiple fields. For self-supervised learning, raw signals are transformed, and models are optimized with reconstruction or contrastive losses against original signals, where preserving of temporal or spatial data consistency is assumed for learning meaningful representations. These representations are proven useful to generalize and solve downstream tasks. On the other hand, multi-task learning [13] improves generality by solving multiple tasks altogether during training, while weighting mechanisms among the losses from each task are crucial [14, 15]. Self-supervised and multi-task learning techniques are combined and applied to the speech domain, and they have shown success in [16, 17], where reconstruction of various hand-crafted features are used for pre-training, and further learned representations are evaluated with downstream emotion recognition and automatic speech recognition (ASR) tasks.

Similar to speech, music is also a highly structured audio signal. There are many hand-crafted features designed specifically for music to solve various music information retrieval (MIR) tasks. In this paper, we are interested in applying self-supervised and multi-task learning methods for pre-training music encoders. We explore various design choices including encoder architectures, weighting mechanisms to combine losses from pretext tasks, and worker selections to reconstruct various music specific hand-crafted features, such as Mel-frequency cepstral coefficients (MFCCs) for timbre [18], Chroma for harmonic [19], and Tempogram [20] for rhythmic attributes. Our main contributions are 1. provide suggestions on best design choice among all the variations from our experiments, and 2. investigate how different selections of pretext tasks interact with the performance of downstream music classification tasks, including instrument, rhythm and genre.

2. METHOD

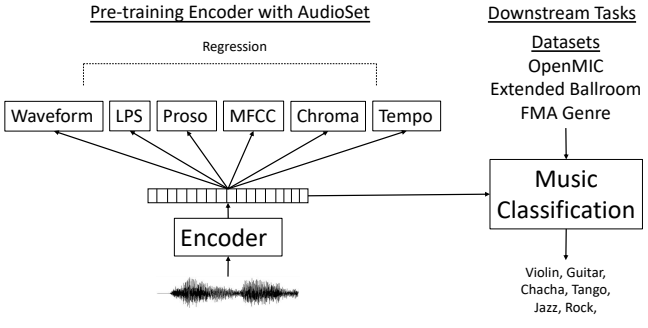

Fig. 1. Diagram of multi-task self-supervised encoder pre-training and downstream music classification evaluation.

⋆ Work done at Amazon

This work is partially supported by the National Science Foundation award #1544753

A two-stage approach involving unsupervised or self-supervised pre-training and supervised learning for training to evaluate on downstream tasks is commonly adopted [9, 10, 16, 17] in recent literature, especially in the context of limited labeled data, where representation learning is key. In order to evaluate the effectiveness of the pre-training, simple linear or multi-layer perceptron (MLP) classifiers are usually used where the pre-trained encoders are required to capture meaningful representations to perform well on linear separation evaluation tasks.

2.1. Multi-task self-supervised pre-training

As shown in Figure 1, we combine self-supervised and multi-task learning ideas for pre-training. Raw audio inputs are passed through multiple encoding layers, and outputs are two dimensional representations with temporal information. These encoded representations are then used for solving pretext tasks via workers including waveform reconstruction, and prediction of various popular hand-crafted features used in MIR to guide the learning jointly.

2.2. Downstream task training scenarios

After pre-training, we remove the workers, and feed the encoder outputs to MLP classifiers for downstream tasks. We adopt three training scenarios proposed in [16]: 1. **Supervised**: Initialize the encoder weights randomly and train from scratch on the downstream datasets directly. 2. **Frozen**: Treat the pre-trained encoder as feature extractor with frozen weights, concatenate the feature extractor with trainable MLP classifiers and only optimize the classifier weights. 3. **Fine-tuned**: Initialize the encoder with pre-trained weights and fine-tune the encoder with downstream tasks altogether.

3. EXPERIMENTAL DESIGN

We experiment with various design choices during pre-training including 1. Encoder architectures, 2. Pretext tasks for worker selections, 3. Weighting mechanisms for losses from pretext tasks. We provide more details on the downstream evaluations and data usage for both pre-training and downstream tasks in section 3.4 and 3.5.

3.1. Encoder architectures

We compare two encoder architectures proposed in two relevant studies in speech domain which inspire our work. We refer the two encoder architectures as PASE [16] and PASE+ [17], respectively.

1. **PASE**: We use the same encoder architecture as the original PASE work [16] with source code implementation[1]. The first layer is based on SincNet [21], where the raw input waveform is convolved with a set of parameterized Sinc functions implementing rectangular band-pass filters. The authors claim that SincNet has fewer parameters and provides better interpretability. SincNet layer is followed by 7 one-dimensional convolutional blocks, batch normalization [22], and multi-parametric rectified linear unit activation [23]. We use the same model parameters as provided in the original work including kernel widths, number of filters, and strides. The set of parameters for convolutional layers emulates a 10ms sliding window.

2. **PASE+**: PASE+ [17] improves upon PASE [16] by adding skip connections and Quasi-Recurrent Neural Network (QRNN) [24] layers to capture longer-term contextual information. QRNN layers consist of interleaved convolutional layers with RNN layers to speed up training with parallel optimization, while maintaining compatible performance.

3.2. Pretext tasks worker selections

Inspired by the original PASE [16] work, we select waveform reconstruction, log power spectrum (LPS) and prosody features as baseline workers. We then choose three popular hand-crafted features in MIR field including MFCC, Chroma, and Tempogram as mixed-in workers. For waveform reconstruction, encoder layers are applied in reverse order to decode embeddings and optimized with mean absolute error (MAE) loss. For all the other workers, we use MLP with convolutional layers, and mean squared error (MSE) loss.

Waveform, LPS, and MFCC are commonly used in machine listening. Chroma is inspired from western 12-tone theory which frequencies are folded into 12 bins as one octave. Tempogram [20] takes local auto-correlation of the onset strength envelope. As used in [16], prosody features include zero crossing rate (ZCR), energy, voice/unvoice probability and fundamental frequency (F0) estimation, resulting in 4 features concatenated along with temporal dimension. For LPS, MFCC, Chroma, Tempogram and prosody, we use librosa[2] implementations with hop_length = 160, n_fft = 2048, sr = 16000 in order to align each hop as 10ms to match encoder parameters, with other default parameters.

3.3. Weighting mechanisms

We explore two weighting mechanisms to combine losses from each worker during pre-training. 1. **Equal weighted** by simply sum up losses from different workers for backpropagation. 2. **Re-weighted** by taking the validation losses per worker of the first 10 epochs from equal weighted training, averaging the loss per worker, taking the reciprocal as the new weights and applying those to retrain from scratch. The intuition is that the losses from each worker will then contribute more equally during backpropagation optimization.

3.4. Downstream evaluation

After pre-training, we remove the workers for pretext tasks and concatenate the output of the encoder with a simple MLP classifier. The input layer of the MLP is to take mean pooling across temporal dimension, resulting in one 512 dimension embedding, followed by 1 fully connected layer to adapt to output dimensions corresponding to the number of classes of each downstream dataset. We train with three scenarios discussed in section 2.2, including supervised, frozen and fine-tuned, all with the same hyper-parameters, Adam optimizer [25] with initial learning rate as 0.001 and early stopping criteria with patience value of 10 on validation loss. We run 10 trials for each experiment in this paper to get statistically meaningful results.

3.5. Data

3.5.1. AudioSet for pre-training

We use clips in AudioSet [26] with "Music" label for pre-training. We are able to acquire ˜2M (97% of the original AudioSet data) clips, within which there are ˜980k clips labeled with "Music". We randomly select 100k for pre-training, resulting in ˜83 hours of data.

[1] https://github.com/santi-pdp/pase

[2] https://github.com/librosa/librosa

3.5.2. Datasets for downstream evaluation

OpenMIC [27], Extended Ballroom [28] and FMA Small (FMA) [29], three publicly available classification datasets are used for downstream evaluation as representative samples of well-known MIR tasks. These datasets range from different number of clips, clip duration, and number of classes. For all three datasets, we report macro F1 scores as shown in the figures.

1. OpenMIC [27]: OpenMIC is a multi-label instrument classification dataset containing 15k samples total with provided train/valid/test splits as well as masks for strong positive and negative examples for each class. We follow similar setup as the official baseline[3] by training 20 binary classifiers.
2. Extended Ballroom [28]: Extended Ballroom (4k samples) is a multi-class dance genre classification dataset. We follow the same setup as [30] by removing 4 categories due to dataset imbalance, resulting in only using 9 categories.
3. FMA Small [29]: FMA Small (8k samples) is a multi-class music genre classification dataset with 8 genre categories.

4. RESULTS AND DISCUSSIONS

We first show results of encoder choices and whether pre-training helps. All workers (waveform (W), LPS (L), prosody (P), MFCC (M), Chroma (C) and Tempogram (T), where WLP are also referred to as baseline) and frozen scenario are used. We then dive deeper into the effects of different weighting mechanisms, and ablation study of worker selections, for which we also report results in frozen scenario. Finally, we investigate whether fine-tuning further improves performance.

4.1. Encoder architectures

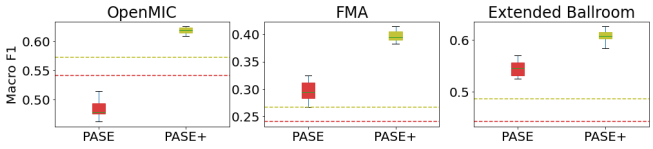

Fig. 2. Comparisons of encoder architectures (PASE vs PASE+). Left, center, and right figures are Macro F1 metrics on different downstream tasks of frozen scenario. Red and green dotted lines represent PASE and PASE+ encoder with supervised training (scenario 1) directly on downstream dataset from scratch.

From Figure 2, we observe that for all three downstream tasks, PASE+ outperforms PASE. This is not surprising as PASE+ is a more powerful encoder with ˜8M parameters, skip-connection and QRNN layer, and PASE has only ˜6M parameters and basic convolutional layers. This confirms with the findings from original PASE+ [17] work applied to speech data.

The dotted lines are trained supervisedly (scenario 1) from scratch directly on the downstream tasks with random weights initialization. It shows that pre-training in general helps to initialize the encoder weights better, resulting in better performance on downstream tasks. One exception is PASE for OpenMIC, we hypothesize that it is because OpenMIC already contains enough data to train PASE encoder (with less capacity) from scratch well, which is not the case for PASE+. This shows that pre-training for encoders with larger capacities is especially helpful when evaluating on downstream tasks with limited labeled data. We conducted experiments using PASE+ through out the remaining paper as it's a better encoder for our tasks.

[3]https://github.com/cosmir/openmic-2018

4.2. Weighting mechanisms

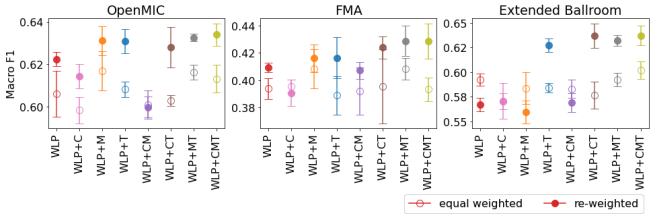

Fig. 3. Comparisons of equal weighted vs re-weighted for different worker selections on all downstream tasks. PASE+ encoder architecture is used with frozen scenarios. Y-axis is Macro F1 classification metrics. X-axis are labeled with WLP (waveform, LPS, and prosody), M (MFCC), C (Chroma), and T (Tempogram). No filled and filled color represent equal weighted and re-weighted mechanisms correspondingly. From all trials, circles represent mean while the length of the bar represents standard deviation.

In Figure 3, we show results comparing equal weighted and re-weighted mechanisms with different worker selections during pre-training. We see that re-weighted mechanism (filled color) helps to boost the influences from various workers to the performance of downstream tasks in general. For Extended Ballroom on the right especially, we see clearly that results with workers containing Tempogram are improved by a large margin.

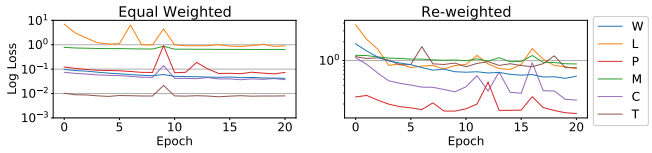

Fig. 4. Log loss per worker for first 20 epochs. X-axis is number of epochs. On the left is equal weighted. On the right is re-weighted where loss weights are balanced using reciprocal of mean losses per worker from equal weighted pre-training.

We further examine losses per worker during pre-training as shown in Figure 4. We can see that with equal weighted on the left, LPS (L) almost dominates all losses and Tempogram (T) worker loss contributes the least with two orders of magnitude smaller, but for re-weighted on the right, each worker contributes more equally.

4.3. Pretext tasks worker selections

Figure 5 shows the relative difference in accuracy by including different workers over the WLP baseline. We observe that different worker selections affect variously to different downstream tasks. Tempogram helps the most across all different combinations especially for Extended Ballroom. MFCC is usually important for most of the downstream tasks as it captures the low-level attributes differentiating instrument and genre. Chroma is however at a disadvantage, especially for OpenMIC, since Chroma is designed to

normalize for timbre, which is important for instrumentation. MFCC only hurts slightly on Extended Ballroom as it brings together different dance genres with similar timbre, and separates music from same dance genre that changes in timbre.

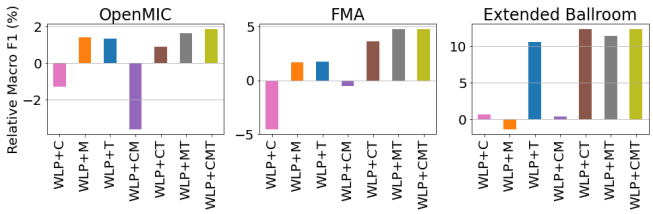

Fig. 5. Relative improvement (%) of different additional music specific workers included during pre-training compared to WLP on different downstream tasks.

These variations can be further compensated to show improvement across all tasks by using all workers as shown on the right most of each subplot in Figure 5. We observe relative improvement adding all workers compared to WLP baseline by 1.9%, 4.5% and 14% on OpenMIC, FMA and Extended Ballroom datasets respectively. This indicates that workers complement each other, and the encoders are able to use signals from diversified workers to generalize better to various downstream tasks.

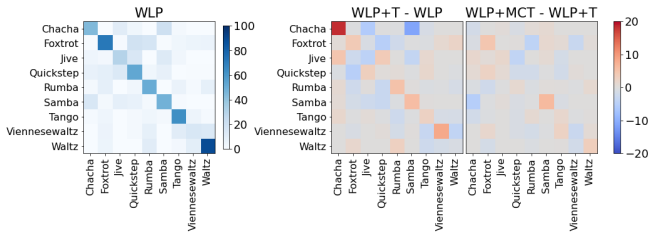

Fig. 6. Confusion matrices of Extended Ballroom. On the left is WLP baseline. On the right are the differences between WLP+T and WLP, and WLP+MCT and WLP+T. Red and blue colors indicate positive and negative changes respectively.

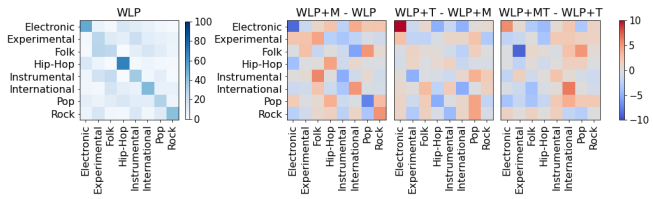

Fig. 7. Confusion matrices of FMA. On the left is WLP baseline. On the right are the differences between WLP+M and WLP, WLP+T and WLP+M, and WLP+MT and WLP+T. Red and blue colors indicate positive and negative changes respectively.

We then show confusion matrices of Extended Ballroom and FMA in Figure 6 and 7. In Figure 6, we show the difference between WLP + T and WLP, and observe that adding Tempogram helps differentiate Chacha with Jive and Samba, which differ in rhythm and tempo, as well as Foxtrot with Quickstep, and Viennesewaltz with Waltz, as the two pairs of dance genres originate from similar music playing in different speed. Adding MFCC and Chroma further helps differentiate Foxtrot with Rumba and Viennesewaltz as additional timbre cues are provided.

In Figure 7, we observe that even adding MFCC (WLP+M - WLP) helps in general as hypothesized, however, it misclassifies Electronic with Hip-Hop and International, and Pop with Hip-Hop and Rock, as there might be similar instruments used in these genres, resulting in similar timbre. Adding Tempogram (WLP+T - WLP+M) corrects the mistakes made on Electronic and Pop genres, but misclassifying International with Folk and Instrumental. Finally, adding both workers (WLP+MT - WLP+T) provides further improvements upon MFCC and Tempogram only. In general we observe improvements with positive values (red) in diagonal and negative (blue) in off-diagonal.

4.4. Frozen versus fine-tuned

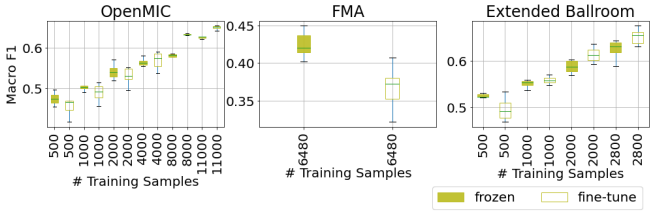

Fig. 8. Comparisons of frozen and fine-tuned on # of training samples for different downstream tasks.

In Figure 8, we plot frozen (filled) versus fine-tuned (no filled) with re-weighted mechanisms and all workers used during pre-training. By using all available training examples, both Extended Ballroom (2.8k) and OpenMIC (11k) show further improvement with fine-tuning, while FMA does not. We hypothesize that this is because each downstream task requires different number of samples for fine-tuned to work well. For FMA, we just do not have enough training samples. We further reduce number of samples used for training OpenMIC and Extended Ballroom as shown in Figure 8, where we see clear reverting behavior around 8k (OpenMIC) and 1k (Extended Ballroom) that fine-tuning stops to outperform frozen.

5. CONCLUSION

In this paper, we explore different design choices for pre-training music encoders with multi-task and self-supervised learning techniques, and show that this method, when combined with different encoder architectures, generally benefits for downstream tasks. The improvement is clearer and more stable when (# unlabeled data / # labeled data) is larger. We also show that each type of pretext task provides different and complementary information, re-weighted mechanism helps the encoder to better learn different cues provided from each task, and fine-tuning can further improve performance.

For future work, we are interested in applying this pre-training technique to various encoders, adding more audio specific features, and explore other unsupervised and self-supervised learning ideas such as wav2vec [5] as pretext tasks. We are also interested in including more diverse downstream tasks such as music tagging, and chord recognition (Chroma should be more effective in this task) for evaluation. We think that this pre-training technique can be applied to a large varieties of music encoders and generalize to different downstream music tasks, especially those with limited labeled data.

6. REFERENCES

[1] Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," *ISMIR 2016*, 2016.

[2] Jaehun Kim, Julián Urbano, Cynthia CS Liem, and Alan Hanjalic, "One deep music representation to rule them all? a comparative analysis of different representation learning strategies," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1067–1093, 2020.

[3] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "Transfer learning for music classification and regression tasks," in *ISMIR 2017*. International Society for Music Information Retrieval, 2017, pp. 141–149.

[4] Jan Wülfing and Martin A Riedmiller, "Unsupervised learning of local features for music classification.," in *ISMIR*, 2012, pp. 139–144.

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

[6] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[7] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[8] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, pp. 14910–14921.

[9] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019*. IEEE, 2019, pp. 3852–3856.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton, "Big self-supervised models are strong semi-supervised learners," *arXiv preprint arXiv:2006.10029*, 2020.

[12] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirović, "Spice: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[13] Yun-Ning Hung, Yi-An Chen, and Yi-Hsuan Yang, "Multitask learning for frame-level instrument recognition," in *ICASSP 2019*. IEEE, 2019, pp. 381–385.

[14] Alex Kendall, Yarin Gal, and Roberto Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[15] Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol, "A comparison of loss weighting strategies for multi task learning in deep neural networks," *IEEE Access*, vol. 7, pp. 141627–141632, 2019.

[16] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *INTERSPEECH*, 2019.

[17] 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*. IEEE, 2020, pp. 6989–6993.

[18] Franz De Leon and Kirk Martinez, "Enhancing timbre model using mfcc and its time derivatives for music similarity estimation," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 2005–2009.

[19] Daniel PW Ellis, "Classifying music audio with timbral and chroma features," 2007.

[20] Peter Grosche, Meinard Müller, and Frank Kurth, "Cyclic tempogram—a mid-level tempo representation for musicsignals," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 5522–5525.

[21] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[22] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proc. of ICML*, 2015.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[24] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher, "Quasi-recurrent neural networks," *Proc. of ICLR*, 2017.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP 2017*. IEEE, 2017, pp. 776–780.

[27] Eric Humphrey, Simon Durand, and Brian McFee, "Openmic-2018: An open data-set for multiple instrument recognition.," in *ISMIR*, 2018.

[28] Ugo Marchand and Geoffroy Peeters, "The extended ballroom dataset," *ISMIR Late-breaking Session*, 2016.

[29] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "Fma: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference*, 2017.

[30] Yeonwoo Jeong, Keunwoo Choi, and Hosan Jeong, "Dlr: Toward a deep learned rhythmic representation for music content analysis," *arXiv preprint arXiv:1712.05119*, 2017.