$$q(k \cdot w) = q(w) \tag{1}$$

In the case of equivariance, the representation reflects the transformation applied to the input:

$$q(k \cdot w) = k \cdot q(w) \tag{2}$$

Although the bulk of self-supervised methods developed so far rely on the notion of invariance to transformations of the training samples, we argue here that equivariance can provide a useful learning signal for forcing representations to capture meaningful properties of the input data. Recent works in the computer vision domain show that employing an equivariant objective in addition to more common invariant objective is beneficial [21, 22]. In this scenario, the equivariance is enforced against transformations such as rotation or scale. Similarly, equivariance is starting to be explored as an additional training objective in the video domain [40].

In the musical audio domain, SPICE relies on a composite training objective with an equivariance constraint to learn pitch representations from unlabelled data at its core [23]. SPICE generates two views of each training sample by applying pitch-shifting and aims to learn representations that are equivariant to the pitch of musical audio. SPICE requires strong regularisation to train effectively, which is achieved by adding an extra decoder network (discarded at inference time) and corresponding audio input reconstruction term to the loss.

In contrast with previous works, the method we propose here solely relies on a simple equivariance loss term. It does not include an invariance-based objective [21, 22] or any extra regularisation loss term [23].

3. METHODS

3.1 Model

Our model pipeline is a close adaptation of the Temporal Convolutional Network (TCN) architecture architecture introduced first in [41] for beat tracking and later extended to joint beat tracking and tempo estimation in [42]. From the audio downmixed to mono, we compute a magnitude spectrogram with a window and FFT size of 2048 samples and a hop size of 441 samples (i.e. 100 frames per second for audio sampled at 44100Hz). The FFT magnitude spectrogram is then mapped to a Mel Spectrogram with 81 bins ranging from 30 to 17,000Hz, and finally logarithmic compression is applied.

The Log Mel Spectrogram is then fed to a neural network constituted of two main building blocks and which architecture is depicted in Figure 1. The input is first fed through a batch normalisation layer followed by 3 convolutional blocks. It is then processed through 8 TCN layers with 16 filters each and geometrically increasing dilation rates from 2^0 to 2^7. Because we focus on learning tempo representations in this work, we drop the beat tracking branch and adapt the tempo branch architecture so that our network outputs a 16-d vector representation $\mathbf{h}$. All other design parameters are identical to [42]. We chose this TCN architecture because it is specifically designed to model temporal characteristics of music and has been shown to provide very competitive performance despite using very little parameters (33k).

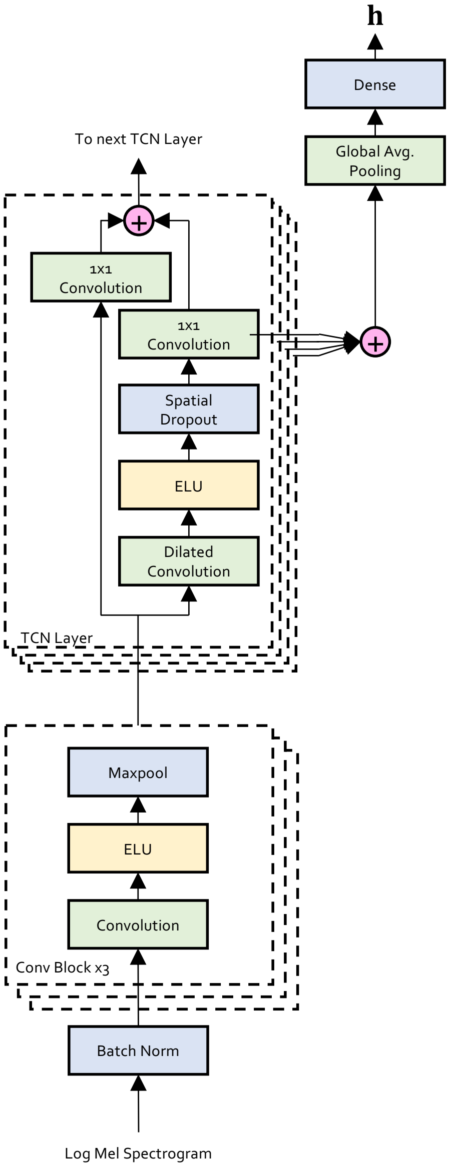

Figure 1. Temporal convolutional neural network (TCN) architecture. The Log Mel spectrogram is first processed through 3 convolution blocks and then through 8 TCN layers with 16 filters and geometrically increasing dilation rates from 2^0 to 2^7. The network outputs a 16-d vector embedding $\mathbf{h}$.

3.2 Training strategy

The objective of the training strategy described below is to learn representations that capture tempo information without having access to tempo annotations at training time. Starting from the observation that the time-stretching modifies the tempo of a music piece, we propose to use equivariance to time-stretching as a self-supervision objective to learn tempo representations.

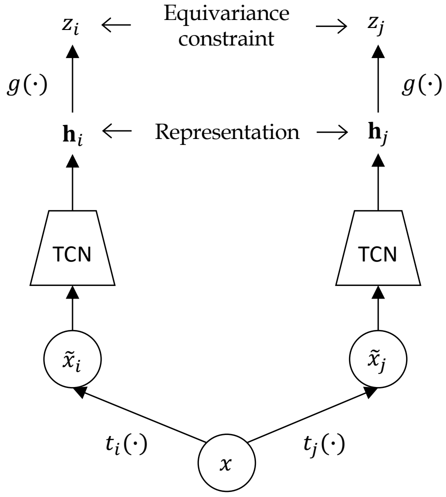

Figure 2. Equivariant self-supervision framework. Two distinct time-stretching transformations (t_i and t_j) are applied to a training sample x to obtain to correlated views ($\tilde{x}_i$ and $\tilde{x}_j$). The TCN network and projection head $g(\cdot)$ are trained to produce a pseudo-tempo scalar z that is equivariant to the time stretching transformation of the input. The projection head is discarded after training.

For a recording taken from the training set, let x be an excerpt of length l_x selected at random in the recording, with unknown tempo $y \in \mathbb{R}$. For each sample x, two views $(\tilde{x}_i, y_i)$ and $(\tilde{x}_j, y_j)$ are produced by applying time stretching transformations noted t_i and t_j respectively using Sox[2]. For each view, the time stretching rate α is drawn uniformly at random from $[1 - r, 1 + r]$ where r is a hyper-parameter to the training procedure. As a result, we obtain:

$$\tilde{x}_i = t_i(x), \qquad y_i = \alpha_i \cdot y \tag{3}$$

$$\tilde{x}_j = t_j(x), \qquad y_j = \alpha_j \cdot y \tag{4}$$

where the right hand side of eq. (3) and eq. (4) materialises the transformation of the tempo of each view.

In order to allow efficient batch processing at training time, we force the augmented views $\tilde{x}_i$ and $\tilde{x}_j$ to all have the same length l_x by cropping if they are longer and padding with zeros if shorter. In all our experiments we set l_x to be 600,000 audio samples (13.6s at 44.1 kHz).

Because the true tempo y is unknown, we propose to use a pseudo-tempo representation $z \in \mathbb{R}$ as the output of model at training time. Let $f(\cdot)$ be the transformation applied by the TCN and $g(\cdot)$ be a linear projection head, so that $g(f(x)) = z$. The objective is then to constrain the TCN and projection head stack to be equivariant to the time-stretching transformation of the input, so that:

$$g\left(f\left(t(x)\right)\right) = \alpha \cdot z \tag{5}$$

Since the two views are derived from the same training sample, it is trivial to show that the equivariance formulation expressed in eq. (5) yields:

$$\alpha_i \cdot z_j = \alpha_j \cdot z_i \tag{6}$$

In other words, the equivariance objective is met if eq. (6) is true. Based on this, we can derive the following training loss that is minimised when the equivariance objective is met:

$$\mathcal{L} = \left| \frac{z_i}{z_j} - \frac{\alpha_i}{\alpha_j} \right| \tag{7}$$

Note how this formulation does not allow the model to collapse on a trivial constant solution to minimise the loss. Producing a constant z value for any input does not yield a minimal loss because α values are drawn at random for every training sample, which means that the ratio $\frac{\alpha_i}{\alpha_j}$ varies for every pair of training sample views.

Other formulations of the loss function that are minimised when eq. (6) is true can be derived, but may allow trivial solutions. For example loss functions such as $\mathcal{L}' = |\alpha_i \cdot z_j - \alpha_j \cdot z_i|$ or $\mathcal{L}'' = \left| z_i - \frac{\alpha_i \cdot z_j}{\alpha_j} \right|$ admit a trivial optimal solution for $z_i = z_j = 0$.

3.3 Training parameters

In all our experiments we use a batch size of 16 samples, the Adam optimiser [43] with initial learning rate of 0.001. We pre-train the model for 20 epochs and fine-tune for 100 epochs.

With the aim to promote robustness against non temporelated attributes of audio signals, we add the following optional random audio augmentations during pre-training: gain, polarity inversion, gaussian noise and SpecAugment frequency masking [44]. Note that our loss function remains unchanged whether or not we apply these augmentations.

3.4 Datasets

For self-supervised training we use the MagnaTagaTune dataset (MTT) [45]. It contains around 25k audio tracks but no tempo annotations.

For fine-tuning and evaluation we use datasets commonly used in the tempo estimation litterature that do contain tempo annotations. Namely we use the following datasets: GTZAN [46], Hainsworth [47] , Giantsteps [48, 49] and ACM Mirum [50].

3.5 Evaluation

After pre-training, we wish to evaluate the representations learnt by the TCN. The projection head $g(\cdot)$ is discarded and all the network weights frozen. We then attach a linear classification head with 300 units and softmax layer to represent the range [0,300] BPM, apply a smoothing window to the ground truth label similar to [42] and fine-tune using the cross-entropy loss.

[2] http://sox.sourceforge.net

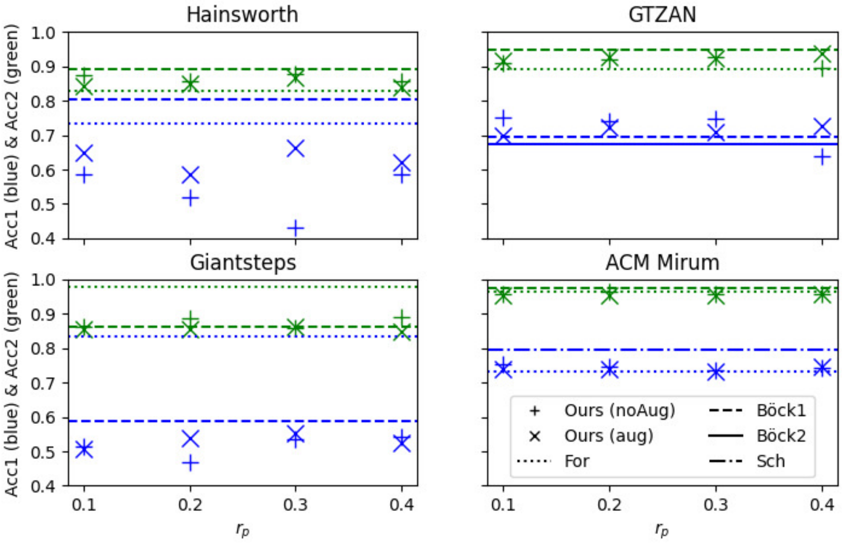

Figure 3. Performance metrics for our proposed method, compared against supervised benchmarks. We report both Accuracy 1 (in blue) and Accuracy 2 (in green). We report results for 8 pre-training conditions, which are combinations of using augmentations ("aug") or not ("noAug") and of $r_p \in \{0.1, 0.2, 0.3, 0.4\}$. In all cases the model is fine-tuned with a time-stretching of strength $r_f = 0.2$. Horizontal lines represent the supervised benchmarks "Böck1" [8], "Böck2" [42], "For" [51], "Sch" [9]. In the interest of legibility, for every dataset and every metric we only display the highest and lowest performing baselines.

In order to evaluate generalisation capability of our model, we perform cross-dataset evaluation. This means that the metrics we report are always computed on a dataset that has never been seen by the model during the pre-training or fine-tuning stages.

In order for our results to be comparable with existing literature, we report tempo performance metrics *Accuracy 1* and *Accuracy 2* scores with a $\pm 4\%$ tolerance [28]. *Accuracy 1* measures the accuracy of the model's prediction of the exact ground truth tempo, while *Accuracy 2* also allows for "octave errors" with factor $\{2, 3, \frac{1}{2}, \frac{1}{3}\}$. We leave further evaluation considerations for future work [52].

4. EXPERIMENTS

4.1 Robustness against trivial solutions

In a preliminary experiment, we ran the pre-training with alternative losses $\mathcal{L}'$ and $\mathcal{L}''$, as defined in Section 3.2. It systematically collapsed to a trivial solution $z \approx 0$. Conversely, we use the loss function $\mathcal{L}$ described in Eq. 7 in all the experiments reported in this paper and did not observe collapse, which confirms its robustness against trivial constant solutions.

4.2 Influence of pre-training augmentation parameters

Figure 3 shows the performance metrics on evaluation datasets for supervised baselines and our method after pre-training on the MTT dataset and cross-dataset fine-tuning and evaluation. We report results for 8 pre-training conditions, which are combinations of using audio augmentations ("aug") or not ("noAug"), and of the strength of time-stretching during pre-training $r_p \in \{0.1, 0.2, 0.3, 0.4\}$. In all cases the model is fine-tuned with a time-stretching of strength $r_f = 0.2$ (see section 4.3).

It appears that adding augmentations yields no notable performance benefit except on Acc1 on the Hainsworth dataset. We hypothesise this is because the TCN architecture already has a strong inductive bias towards temporal and rhythmic structures, which makes it robust against potentially confounding attributes of the audio signal.

The choice and strength of input transformations has been shown to have an impact on the performance of contrastive learning [20]. Intuitively, one expects that very gentle augmentations may not allow to learn robust representations, on the other hand too strong an augmentation may lose semantic content and therefore not allow to learn at all. In our experiments, we observe that the performance tends to dip slightly at the extreme ends of the range of values for r_p. However, it is interesting to note that we do not observe a dramatic degradation of performance even though the transformations applied then (e.g. $r_p = 0.4$) are musically extreme.

Overall these results suggest that the pre-training phase is fairly robust to the choice of time-stretching parameters and to the absence of audio augmentation.

4.3 Influence of fine-tuning augmentation parameters

At the fine-tuning stage, tempo estimation is formulated as a multiclass classification problem, where each class corresponds to a tempo value. Because it is likely that the datasets used for fine-tuning may not contain training sam-

Method	r_f	Hainsworth		GTZAN		Giantsteps		ACM Mirum	
		Acc1	Acc2	Acc1	Acc2	Acc1	Acc2	Acc1	Acc2
Ours	0.0	0.604	*0.838*	*0.691*	0.887	0.512	0.809	0.704	0.943
Ours	0.1	0.586	0.824	*0.719*	0.881	0.456	0.791	*0.757*	*0.965*
Ours	0.2	0.518	*0.856*	*0.741*	*0.919*	0.470	*0.886*	*0.747*	*0.965*
Ours	0.3	0.550	*0.829*	**0.785**	*0.921*	0.438	0.846	0.700	0.958
Ours	0.4	0.541	*0.829*	*0.778*	*0.926*	0.472	*0.884*	0.724	0.952
Schreiber [9]	-	0.770	0.842	0.694	0.926	0.730	0.893	**0.795**	0.974
Foroughmand [51]	-	0.734	0.829	0.697	0.891	**0.836**	**0.979**	0.733	0.965
Böck 1 [8]	-	**0.806**	**0.892**	0.697	**0.950**	0.589	0.864	0.741	**0.976**
Böck 2 [42]	-	-	-	0.673	0.938	0.764	0.958	0.749	0.974

Table 1. Influence of time stretching parameters during Fine-tuning, and comparison to supervised baselines. In all cases, our model is pre-trained on MTT with $r_p = 0.2$ and no audio augmentations. Highest performance for each metric and dataset shown in bold. Metrics where our model outperforms at least one of the baselines in italics.

ples for each tempo in the full BPM range considered, we also apply a time-stretching augmentation to increase the range of tempi seen during fine tuning.

Table 1 summarises the results using a model pre-trained with $r_p = 0.2$ and no invariant augmentations, for a range of fine-tuning time-streching augmentation strength r_f. As expected, it appears that applying some time-stretching ($r_f > 0$) generally improves performance over not applying any ($r_f = 0$). We also note that the optimal value varies from one dataset to the next. For example results seem to be optimal for relatively large augmentation strength on the GTZAN dataset while they would be optimal for smaller values on ACM Mirum.

4.4 Comparison to supervised benchmarks

Figure 3 shows supervised benchmarks as horizontal lines. Our proposed method's Accuracy 2 performance is comparable with supervised benchmarks on all datasets. Accuracy 1 performance is more inconsistent. It lags behind supervised methods on Hainsworth and Giantsteps datasets, while it is comparable with supervised benchmarks on ACM Mirum. Notably, our method outperforms all supervised baselines in Accuracy 1 on GTZAN. Similar conclusions can be drawn from the results shown in Table 1, under different fine-tuning configurations.

Note that we evaluate our models on datasets that have never been seen during pre-training or fine-tuning and still observe performance generally competitive with supervised benchmarks. Also taking into account that we only fine-tune a linear layer, this result indicates a promising degree of robustness of the representation learnt during pre-training against domain shift.

5. CLOSING WORDS

In this work, we introduced an approach to use equivariance as a self-supervision signal to learn audio tempo representations from unlabelled data. We derive a simple loss function that prevents the network from collapsing on a trivial solution during training, without requiring any form of regularisation or negative sampling. Our experiments show not only that it is possible to learn meaningful representations for tempo estimation by solely relying on equivariant self-supervision, but also demonstrate that we can achieve performance comparable with supervised methods on most benchmarks. We also show that the representations learnt exhibit promising robustness against pre-training and fine-tuning hyper-parameters as well as against domain shift. As an added benefit, our method only requires moderate compute resources by making use of a small model and not requiring large batch sizes, therefore keeping it accessible to a wide research community.

We believe these results open a number of interesting avenues for future work. This paper is focused on tempo estimation but there is potential to extend our investigation to other MIR tasks revolving around rhythmic properties such as beat tracking, metrical structure estimation or rhythm pattern identification. In addition, because no annotated data is required for pre-training, there is potential for investigating applications to low resource musical genres or genres underserved by traditional and current supervised methods. Since our loss function is very simple, it could also be added as an extra objective in SSL frameworks for learning general music representations, at a moderate cost of increased complexity. Last but not least, while invariance typically used in contrastive learning is naturally connected with classification problems, equivariant self-supervision is well suited to tasks that can be formulated as regression problems. We therefore believe there is potential to explore many more applications of equivariant SSL in the music domain and beyond.

6. REFERENCES

[1] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"

arXiv:1810.04805, Oct. 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," *arXiv:2002.05709*, Feb. 2020. [Online]. Available: http://arxiv.org/abs/2002.05709

[4] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[5] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.

[6] D. Yao, Z. Zhao, S. Zhang, J. Zhu, Y. Zhu, R. Zhang, and X. He, "Contrastive Learning with Positive-Negative Frame Mask for Music Representation," *arXiv preprint arXiv:2203.09129*, 2022.

[7] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, "S3T: Self-Supervised Pre-training with Swin Transformer for Music Classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 606–610.

[8] S. Böck, F. Krebs, and G. Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *International Society for Music Information Retrieval (ISMIR) Conference*, 2015.

[9] H. Schreiber and M. Müller, "A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network." in *International Society for Mu- sic Information Retrieval (ISMIR) Conference*, 2018, pp. 98–105.

[10] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 12–16.

[11] X. Sun, Q. He, Y. Gao, and W. Li, "Musical Tempo Estimation Using a Multi-scale Network," *arXiv preprint arXiv:2109.01607*, 2021.

[12] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[13] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[14] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[15] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006, pp. 1735–1742.

[16] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, "Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning," *arXiv preprint arXiv:2010.13991*, 2020.

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

[18] P. Manocha, Z. Jin, R. Zhang, and A. Finkelstein, "CDPAM: Contrastive learning for perceptual audio similarity," *arXiv preprint arXiv:2102.05109*, 2021.

[19] S. Srivastava, Y. Wang, A. Tjandra, A. Kumar, C. Liu, K. Singh, and Y. Saraf, "Conformer-Based Self-Supervised Learning for Non-Speech Audio Tasks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[20] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 6827–6839, 2020.

[21] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljačić, "Equivariant Contrastive Learning," *arXiv preprint arXiv:2111.00899*, 2021.

[22] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 275–12 284.

[23] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[24] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, p. 588, 1998.

[25] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram Representation and Kalman filtering," *Journal of New Music Research*, vol. 29, no. 4, pp. 259–273, 2000.

[26] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[27] M. Alonso, B. David, and G. Richard, “A study of tempo tracking algorithms from polyphonic music signals,” in *Proceedings of the 4th. COST 276 Workshop*, 2003.

[28] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, “An experimental comparison of audio tempo induction algorithms,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[29] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, “A tutorial on onset detection in music signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[30] S. Böck and G. Widmer, “Maximum filter vibrato suppression for onset detection,” in *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx)*, 2013.

[31] M. E. Davies and M. D. Plumbley, “Causal Tempo Tracking of Audio.” in *International Society for Music Information Retrieval (ISMIR) Conference*, 2004.

[32] G. Peeters, “Time variable tempo detection and beat marking,” in *Proceedings of the ICMC*, 2005.

[33] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[34] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, and M. Gheshlaghi Azar, “Bootstrap your own latent-a new approach to self-supervised learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[35] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “Byol for audio: Self-supervised learning for general-purpose audio representation,” in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[36] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” *arXiv preprint arXiv:2103.03230*, 2021.

[37] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[38] X. Chen and K. He, “Exploring Simple Siamese Representation Learning,” *arXiv preprint arXiv:2011.10566*, 2020.

[39] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[40] S. Jenni and H. Jin, “Time-equivariant contrastive video representation learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9970–9980.

[41] E. P. MatthewDavies and S. Böck, “Temporal convolutional networks for musical audio beat tracking,” in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[42] S. Böck, M. E. Davies, and P. Knees, “Multi-task learning of tempo and beat: learning one to improve the other,” in *20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019.

[43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[44] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

[45] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of Algorithms Using Games: The Case of Music Tagging.” in *ISMIR*, 2009, pp. 387–392.

[46] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[47] S. W. Hainsworth and M. D. Macleod, “Particle filtering applied to musical tempo tracking,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2385–2395, 2004.

[48] P. Knees, A. Faraldo Perez, H. Boyer, R. Vogl, S. Böck, F. Horschlager, and M. Le Goff, “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[49] H. Schreiber and M. Müller, “A Crowdsourced Experiment for Tempo Estimation of Electronic Dance Music.” in *ISMIR*, 2018, pp. 409–415.

[50] G. Peeters and J. Flocon-Cholet, “Perceptual tempo estimation using GMM-regression,” in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012, pp. 45–50.

[51] H. Foroughmand and G. Peeters, “Deep-rhythm for tempo estimation and rhythm pattern recognition,” in *International Society for Music Information Retrieval (ISMIR)*, 2019.

[52] H. Schreiber, J. Urbano, and M. Müller, “Music Tempo Estimation: Are We Done Yet?” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

HOW MUSIC FEATURES AND MUSICAL DATA REPRESENTATIONS AFFECT OBJECTIVE EVALUATION OF MUSIC COMPOSITION: A REVIEW OF THE CSMT DATA CHALLENGE 2020

Yuqiang Li[1] **Shengchen Li**[1] **George Fazekas**[2]

[1]Xi'an Jiaotong-Liverpool University
[2]Queen Mary University of London
{yuqiang.li19@student., shengchen.li@}xjtlu.edu.cn,
g.fazekas@qmul.ac.uk

ABSTRACT

Tools and methodologies for distinguishing computer-generated melodies from human-composed melodies have a broad range of applications from detecting copyright infringement through the evaluation of generative music systems to facilitating transparent and explainable AI. This paper reviews a data challenge on distinguishing computer-generated melodies from human-composed melodies held in association with the Conference on Sound and Music Technology (CSMT) in 2020. An investigation of the submitted systems and the results are presented first. Besides the structure of the proposed models, the paper investigates two important factors that were identified as contributors to good model performance: the specific music features and the music representation used. Through an analysis of the submissions, important melody-related music features have been identified. Encoding or representation of the music in the context of neural network modes are found noticeably impacting system performance through an experiment where the top-ranked system was re-implemented with different input representations for comparison purposes. Besides demonstrating the feasibility of developing an objective music composition evaluation system, the investigation presented in this paper also reveals some important limitations of current music composition systems opening opportunities for future work in the community.

1. INTRODUCTION

With the rapid development of AI, automatic music composition systems are considered to approach the quality of human-composed melodies in their output under certain conditions. The Conference on Sound and Music Technology (CSMT) organised a data challenge in 2020 to investigate how to tell apart human-composed melodies from computer-generated melodies, where participants were required to develop an algorithm (or a system) that can distinguish computer-generated melodies from human-composed ones. The submitted algorithms can potentially be used to prevent computer systems from being accused of music copyright infringement, and to objectively measure the performance of different algorithmic composition systems.

The primary task in this data challenge was to develop a system to identify human-composed melodies in a dataset where they are mixed with computer-generated melodies. The generated melodies come from several state-of-the-art music generation frameworks [1], including Variational Auto-Encoder (VAE) [2], Transformer [3] and Generative Adversarial Network (GAN) [4]. A training set with only generated melodies was released first, followed by an evaluation set with human-composed melodies mixed in. Two datasets were used for training the music generation systems separately: Bach Chorales [1] in Music21 [5] and HookTheory [2]. The generated and composed melodies thus followed two styles associated with Bach and more generally the pop genre.

The rankings of the participants were determined by the AUC score in ROC tests, based on the task of identifying human-composed melodies. 7 teams participated in the data challenge with a total of 14 submitted systems, covering various types of algorithms. For instance, rule-based system [6], LSTM (Long short-term memory) [7–9], AE (auto-encoder) [10] and more conventional SVM (support vector machine) [11]. The top-ranked submission by Li *et. al* [7] obtained an AUC score of 0.88, which demonstrated the feasibility of using a computer to identify the music source, at least in a specific context of the two styles by distinguishing human compositions from generated melodies.

The results of the data challenge are further investigated in three ways: style, pitch feature and music representation. The sub-rankings regarding the two music styles are also compared, though little difference was observed. However, the system performance shows significant differences in terms of different pitch features in melodies and music representation methods. Moreover, the paper also presents a revised version of the top-ranked system with

Attribution: Y. Li, S. Li, and G. Fazekas, "How Music Features and Musical Data Representations Affect Objective Evaluation of Music Composition: A Review of the CSMT Data Challenge 2020", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

[1] https://web.mit.edu/music21/doc/moduleReference/moduleCorpusChorales.html

[2] https://www.hooktheory.com/

different input representations to discuss how music representation affects the identification of melodies.

The identification of computer-generated melodies could be used to ensure that no copyright is claimed for fully-automated computer-generated melodies. [12]. Moreover, measuring similarity between human and computer-generated melodies could become a component of Generative Adversarial Networks (GAN) and similar frameworks for automatic and assistive composition, as well as reduce the subjective bias in the evaluation of music generation systems.

The remainder of the paper is organised as follows: we first review the CSMT 2020 data challenge including submissions and results. The performance differences between submissions regarding different musical features are then investigated. Two experiments on the impacts of music representations are presented followed by a brief conclusion.

2. DATA CHALLENGE

2.1 Task Definition and Organisation

The proposed task for the data challenge was to distinguish human-composed melodies from computer-generated melodies, all in the form of 8-bar single-track MIDI files. A training dataset was released first, containing only computer-generated melodies. The evaluation dataset, with equal numbers of human-composed melodies and generated melodies, was released without the associated labels one month before the final submission deadline. The final results were ranked using the AUC score in ROC test, which identifies human-composed melodies from computer-generated ones.

Time	Event
July 15th, 2020	Release of the development dataset
August 15th, 2020	Release of the evaluation dataset
September 15th, 2020	Deadline of submission (prediction, model and report)
October 20th, 2020	Submission review finished
November 4th, 2020	Result announcement

Table 1: Timeline of the CSMT 2020 Data Challenge

2.2 Dataset

A detailed specification of the dataset, especially the training dataset, can be found in [1]. The components of evaluation set are shown in Table 2 to provide a clearer context of the results presented in this paper. Notice that according to [1], 95% of the human-composed melodies in the evaluation dataset are randomly sampled from those that were used to train the three types of generative models.

2.3 Baseline System

An Auto-Encoder (AE) neural network was used as the baseline system of the challenge, whose source code is available on the official website. Figure 1 plots the model architecture. The model uses a bottleneck structure with the cross entropy loss.

	MTrans	MVAE	MNet	Human	Total
Bach	600	200	200	1000	2000
Pop	600	200	200	1000	2000
Total	1200	400	400	2000	4000

Table 2: The evaluation dataset used in the challenge. "MTrans", "MVAE", "MNet" are short for Music Transformer [13], MusicVAE [2] and MidiNet [4], respectively. Music styles "Bach" and "Pop" are listed separately.

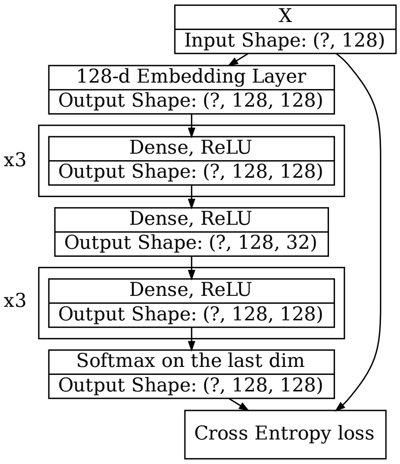
X
Input Shape: (?, 128)
128-d Embedding Layer
Output Shape: (?, 128, 128)
x3
Dense, ReLU
Output Shape: (?, 128, 128)
Dense, ReLU
Output Shape: (?, 128, 32)
x3
Dense, ReLU
Output Shape: (?, 128, 128)
Softmax on the last dim
Output Shape: (?, 128, 128)
Cross Entropy loss

Figure 1: Baseline Auto-Encoder Network Architecture

The baseline model was trained with only real melodies from the Nottingham Folk Tunes dataset[3], consisting of 1,200 American and British songs. The chord information provided in the dataset was not used. The baseline system characterises features of human-composed melodies with an auto-encoder, where computer-generated melodies will result in a high reconstruction loss.

2.4 Overview of the Submitted Systems

For each submitted system, participants were allowed to provide multiple predictions based on different settings or parameters of a model. The challenge received 14 sets of predictions submitted by 7 participants. To simplify references to the submitted systems, aliases are introduced in Table 3 to represent the 8 models. In the table, the presented AUC score is the highest among all predictions of each submitted system.

Around half of the submitted systems outperformed the baseline, these are listed in Table 3. Most participants adopted Deep Neural Networks (DNN), but conventional ML algorithms and rule-based methods were also observed.

There are three main types of methodologies among the submissions: outlier detection, binary classification and heuristics. The outlier detection systems [10] and [14] only learn the feature distribution of the samples of one class

[3] `https://github.com/jukedeck/nottingham-dataset`