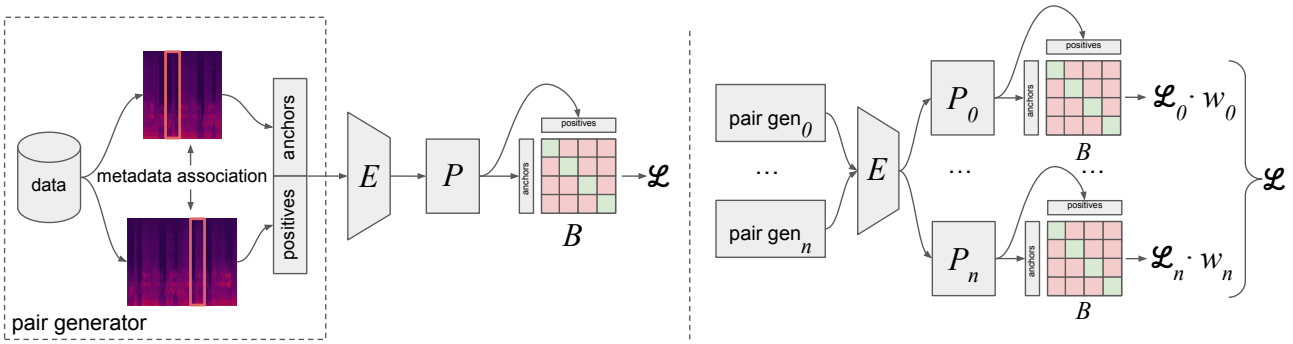

Figure 1. Overview of the proposed pipelines for a single (left) and multiple tasks (right). *E* is the encoder, *P* the projection head, *B* are the weights for the bilinear similarity, and $\mathcal{L}$ is the loss term. In the multi-task setup, tasks go from *0* to *n*.

showed a positive correlation with the downstream metrics, we will only perform full-size experiments and focus on unexplored aspects.

In particular, we are interested in imposing tight constraints on the pair generation process to emphasize the differences between the different associations. To formalize our problem, we consider a collection of music where each song has a track ID (*t*) and appears in one or more releases (*R*). Each release is produced by one or more artists (*A*) for one or more record labels (*L*). We define one self-supervised and three metadata-based associations:

- *track association*, the anchor and positive samples come from the same track (self-supervised),

$$t_a = t_p$$

- *release association*, t_a and t_p have at least a release in common,

$$|R(t_a) \cap R(t_p)| > 0$$

- *artist association*, t_a and t_p have at least an artist in common, but they do not appear in the same release,

$$|A(t_a) \cap A(t_p)| > 0, \text{ and } |R(t_a) \cap R(t_p)| = 0$$

- *label association*, t_a and t_p have at least a record label in common, but they do not share any artist,

$$|L(t_a) \cap L(t_p)| > 0, \text{ and } |A(t_a) \cap A(t_p)| = 0$$

Note that this approach considerably limits the number of pairs that can be matches and presumably leads to sub-optimal representations. However, our goal is to limit the number of overlapping pairs to emphasize the differences between associations. For instance, the release associations would be broadly a subset of the artist ones without the proposed constraints.

For each association, we generate a dataset of anchor/positive pairs. We initialize a pool with all the songs in the music collection. For each song, we pick a random pair that complies with the association's condition and remove both from the pool. While this approach does not exploit every possible association, it gives each track one association attempt, which provides us with sufficient training data for the scope of this work. We also considered a balanced version of the algorithm (e.g., same number of tracks per artist) that we discarded as the performance dropped due to the reduced dataset size without providing additional insights.

Following our contrastive paradigm, we do not need to create explicit anchor/negative pairs. For a given anchor, the rest of the positive samples in the batch are considered negatives (see Figure 1). While this naive approach implies that two pairs of tracks associated with the same artist will be respectively used as negative examples, there is a small probability of having repeated artists in a batch, and we ignored this issue. [2]

4. EXPERIMENTS

We pre-trained different contrastive systems following the proposed similarity notions. To evaluate the quality of the learned representations, we trained shallow classifiers on different multi-class and multi-label classification tasks. Additionally, we conducted experiments to understand the complementarity of such representations.

4.1 Contrastive targets based on Discogs' metadata

The Discogs database provides publicly available dumps of their *release* data that we used to create our training targets. According to Discogs, a release is "a broad term for any audio product that is made for general public consumption"; albums, singles, or compilations are examples of releases. Each release entry contains lists of the artists and record labels involved, the year and country of release, and a list of music genres and styles according to the Discogs' taxonomy. We matched our in-house audio collection to releases and master releases (groupings of different versions of a release) in the Discogs metadata dump resulting in 4 million tracks, with 58.2% of the tracks belonging to more than one release. A track may be linked to many releases for multiple reasons, including remasters, reeditions, or compilations containing it. For each track, we generated lists of artists and record labels by pooling the metadata on the

[2] For example, considering a dataset of one million tracks, a batch size of 200 pairs, and an average of 6 tracks per release, the probability of having two pairs from the same release in the batch is $6e-4$, which we consider an affordable false-negative rate.

Association	Pairs	Diversity	Time	Acc.
track	3.3M	3.3M	63h	88.7
release	846K	2.0M	21h	35.7
artist	1.2M	257K	33h	41.1
label	1.1M	142K	48h	24.7

Table 1. Statistics for the metadata association and their respective models. We show the number of pairs used, association diversity (number of different tracks, releases, artists, and record labels, respectively), training time (hours), and validation accuracy (%).

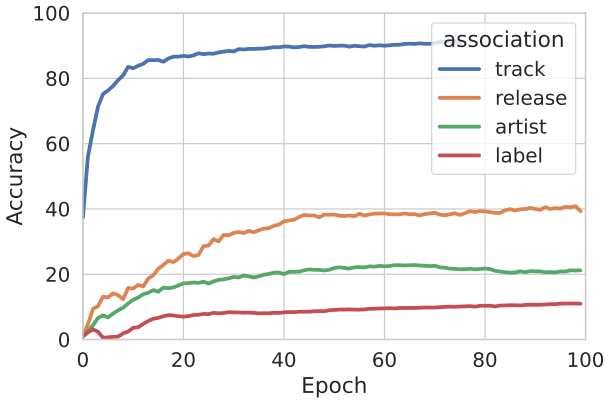

Figure 2. Training accuracies over the epochs for the different associations.

releases linked to it. We observed that different versions of a release could contain very different information, so for us, this was a simple way of maximizing the amount of information available per track.

We selected the subset with the top-400 most popular music styles resulting in 3.3 million tracks to train a baseline model with a multi-label classification objective (Style tags model). We reserved subsets of 50,000 tracks without artist overlap and a minimum frequency of 50 releases per music style tag for testing and validation. These sets were used as data pools to generate training, validation, and testing sets for the considered metadata associations following the methodology presented in Section 3. Note that we did not apply any data cleaning or deduplication, meaning that there may be room for improving the proposed representation models. Table 1 contains the resulting number of pairs and association diversities for each association, as well as the training time, and the accuracy obtained by their respective models. [3]

4.2 Pre-training the embedding models

We used an EfficientNet v1 on its B0 configuration as a backbone CNN to learn the embedding representations [37]. Our models operate on 2-second patches of mel-spectrograms with 96 bands extracted with the same parametrization as for MusiCNN [38] using the Essentia library [39]. [4]

Due to the massive dataset size, we stored the features as half-precision (16-bit) floats, and split the data into three machines with two GeForce RTX 2080 Ti GPUs each to parallelize the training. Every epoch, we fed the model with a random 2-second crop of the mel-spectrogram from each track in the training set. We also used a random patch per track for validation, but in this case, we used the same offset on every epoch to get more stable metrics. We relied on the Adam optimizer with an initial learning rate of $1e{-}3$ and a scheduler that reduced the learning rate by a factor of 10 if the validation loss had not decreased in 10 epochs. We trained the models for 100 epochs and considered two versions. The first one had the weights from the epoch where the lowest validation loss was achieved. The second model was obtained with stochastic weight averaging [40]. For the last 25 epochs, we imposed a learning rate of $1e{-}3$ and kept a moving average of the weights. As the latter version only reported minimum improvements in specific tasks, we decided to present the results of the former only.

As a baseline, we trained a model targeting the top-400 Discogs music styles with the multi-label soft-margin loss by connecting a fully-connected layer to the flattened output of the last convolutional block with 1280 units (embedding layer). As discussed on the Discogs website, [5] music styles usually go beyond purely stylistic descriptions and encode cultural, temporal, or geographical information, so we hypothesize that the learned representations are valuable beyond the task of genre recognition.

For the contrastive objectives, we used a fully-connected projection head with 512 dimensions on top of the same embedding layer, followed by a normalization layer and a *tanh* activation. We used the bilinear similarity ($a^T Bp$) and the cross-entropy loss as in the original implementation. While the authors of COLA showed that larger batch sizes improved the performance, we could only afford a batch size of 200 pairs due to the memory size of our GPUs. We parallelized the training so that each optimizer step aggregated six batches computed by different GPUs. This setup was close to the optimal number of pairs per optimizer step found in the original publication but used a larger ratio of positive samples.

To get additional insights into the models, we computed the top-1 accuracies by taking the *arg max* of the similarity matrices. Table 1 shows the training times and validation accuracies obtained by the model of each metadata association, and Figure 2 shows the evolution of the training accuracies over the epochs. The accuracies show that the associations have different difficulty levels, which aligns with our expectations.

All the pre-trained models are publicly available for feature extraction within the Essentia library. [6]

[3] The high number of releases is partially due to albums with multiple reeditions. On average, each track in our dataset is linked to 5 releases.

[4] `https://essentia.upf.edu/reference/std_TensorflowInputMusiCNN.html`

[5] `https://blog.discogs.com/en/genres-and-styles`

[6] `https://essentia.upf.edu/models.html`

Dataset	Size	Classes	Type
Genre	55,215	87	Multi-label
Instrument	25,135	40	Multi-label
Mood	18,486	56	Multi-label
Top50	54,380	50	Multi-label
MTAT	25,860	50	Multi-label
FMA	8,000	8	Multi-class

Table 2. Considered downstream datasets.

4.3 Downstream datasets

We evaluated our models as frozen feature extractors following a transfer learning setup. We consider three well-known music auto-tagging and classification datasets: FMA-small [41] (*FMA*), MagnaTagATune [42] (*MTAT*), and the MTG-Jamendo Dataset [43]. The latter contains different subsets for *Genre*, *Instrument*, and mood/theme (*Mood*) tags, as well as the top-50 tags in the dataset (*Top50*) that we treated independently. For *FMA* and *MTAT*, we used the splits proposed by the authors [41] and the 12:1:3 partition [44] respectively. In the MTG-Jamendo tasks, we used the sets defined by its *split-0* similarly to previous works [17, 45]. Table 2 shows the size of the considered datasets.

4.4 Transfer learning evaluation setup

As for the pre-training stage, we trained the downstream models with 2-second patches randomly cropped on each epoch. For validation, we averaged over the activations from the half-overlapped patches of the entire tracks. We passed the patches through the frozen backbone and used the flattened output of the last convolution layer as the input to a multi-layer perceptron with a single hidden layer of 512 units with a *sigmoid* or *softmax* activation for the multi-label or multi-class tasks. We used the cross-entropy loss and the Adam optimizer with a starting learning rate of $1e-3$ and added a weight decay of $1e-5$. A scheduler divided the learning rate by half if the loss had not decreased in five epochs. We trained the models for 30 epochs and used the weights from the epoch achieving the lowest validation loss to evaluate the test set.

4.5 Stacks of embeddings

Apart from evaluating the embeddings obtained from every association individually, we were interested in understanding their complementarity. We wanted to understand if the individually-learned representations could be combined to boost the performance, and if so, which were the best combinations. To investigate this, we ran stacks of embeddings through the presented evaluation protocol. First, we evaluated the stack of the Track, Release, Artist, and Label features for all the datasets. Additionally, we performed a systematic evaluation considering all the possible combinations of the four contrastive models plus the model trained on tags on *MTAT* considering two, three, four, and five embeddings.

4.6 Multi-task model

We also considered training a multi-task model to learn the metadata associations jointly. The architecture of the proposed system is depicted in Figure 1 (right). For each association, we have a separate pair generator and projection head. To perform an optimization step, we ran a batch of pairs from each association through the shared encoder and its specific projection head and computed a weighted sum of the losses. The loss weights were empirically selected prioritizing associations with better single-model performances (*track*: 0.1, *release*: 0.15, *artist*: 0.6, and *label*: 0.15). Additionally, we initialized the encoder with the weights of the model based on artist associations on its 20th epoch to speed up the training. While we experimented with multi-task models based on two association types, we did not find additional insights and decided to omit those results. Due to the additional model size, we had to reduce the batch size to 50 pairs per association.

4.7 Results and discussion

Table 3 reports the metrics obtained by our models and selected works from the literature. In descending order, the five groups in the table contain SOTA models trained from scratch without additional data, SOTA embedding models, baseline embedding models trained by us, our proposed embedding models trained on metadata associations, and models combining metadata associations. In the first group, we include MusiCNN [38], Harmonic CNN [45], and the winning submission of the 2021 Emotion and Theme Recognition challenge (team Lileonardo) [46] [7] as the best models from the literature in *MTAT*, *Top50*, and *Mood*, respectively. Similarly, MuLaP [17] reports the best performance as a frozen embedding extractor in *Genre*, *Mood*, *Top50*, and *FMA*, and the same applies to CALM [10] in *MTAT*. Note that comparisons against these reported metrics may be unreliable due to differences in training and evaluation settings.

We computed three baselines based on audio embeddings from an EfficientNet architecture with random weights, an EfficientNet architecture trained on music style tags as described in Section 4.2, and the VGGish model with its pre-trained weights [47].

Concerning our contrastive models, we observe that the model based on track associations (Track model) achieves competitive performance in some tasks, especially in *Top50*. Nevertheless, the models using metadata associations show better or equivalent performance despite seeing fewer pairs of tracks in the pre-training stage. In particular, we find that model based on artist associations (Artist model) is the best-performing with a few exceptions, which aligns with previous studies in metadata-based music representation learning [13].

Instrument and *MTAT* are the tasks where our models are further from the SOTA. For *MTAT*, we attribute this to the fact that CALM has 1,000 times more parameters, and

[7] https://multimediaeval.github.io/2021-Emotion-and-Theme-Recognition-in-Music-Task/

	Genre		Instrument		Mood		Top50		MTAT		FMA
	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	Acc.
Lileonardo	-	-	-	-	**77.5**	**15.1**	-	-	-	-	-
Harmoic CNN	-	-	-	-	-	-	83.2	29.8	*91.3	*45.9	-
MusiCNN	-	-	-	-	-	-	-	-	90.7	38.4	-
MuLaP	85.9	-	76.8	-	76.1	-	82.8	-	*89.3	*40.2	**61.1**
CALM	-	-	-	-	-	-	-	-	**91.5**	**41.4**	-
Random weights	50.7	3.1	49.9	6.4	50.4	3.4	48.3	6.5	50.0	5.3	12.5
Style tags	87.7	19.9	77.6	19.8	75.6	13.6	83.1	29.7	90.2	37.4	59.1
VGGish	86.3	17.2	**77.8**	**20.2**	76.3	14.1	83.2	28.2	90.2	37.2	53.0
Track associations	86.3	18.0	69.9	16.7	74.0	12.8	82.9	29.4	89.7	36.4	58.9
Release associations	86.9	18.9	71.9	17.2	72.8	11.7	83.2	29.8	90.3	37.1	60.9
Artist associations	**87.7**	**20.3**	69.7	16.9	76.3	14.3	**83.6**	**30.6**	90.7	38.0	59.1
Label associations	87.0	19.4	75.0	18.2	74.8	12.8	83.1	29.9	88.7	34.2	59.5
Stack	86.9	19.4	74.7	18.8	74.3	13.0	83.4	30.0	90.8	38.6	59.8
Multi-task	87.2	19.9	70.5	17.2	76.1	14.4	83.5	30.3	90.8	37.8	60.0

Table 3. ROC-AUC, PR-AUC, and accuracy metrics for the downstream datasets. The five horizontal groups represent SOTA models from the literature trained from scratch, SOTA feature extractors from the literature, baseline feature extractors trained by ourselves, the proposed feature extractors based on metadata associations, and the proposed feature extractors combining associations. (*) results were computed in a clean version of *MTAT* and are not directly comparable.

that the authors report the best metrics from a grid search over shallow classifiers and hyperparameters. Also, we observed that models operating in the full *MTAT* previews tend to report higher PR-AUC performances [8, 10] than models operating in short chunks [5, 38].

The Stack is obtained by concatenating the embeddings from the models based on single associations as input for the MLPs. Except for *MTAT*, we do not get improvements over the best single model in any dataset despite the additional input dimensionality (5,120). Table 4 shows the top-5 results of models trained on combinations of embeddings for *MTAT*. The representations from the Artist and Style tags models are the only ones present in all the top-5 combinations, suggesting that these are the representations with the most valuable information. This observation aligns with the results from Table 3.

Similarly, our Multi-task model is not superior to the best single model in any dataset, but generally it is close in performance to the Artist model. We observe that Multi-task only overcomes Artist in the datasets where other associations perform better (i.e., *Instrument* and *FMA*), which shows its capability to pick the best information from different association types.

Tr	Re	Ar	La	St	ROC-AUC	PR-AUC
✓		✓	✓	✓	90.93	38.75
✓	✓	✓		✓	90.92	38.69
	✓	✓	✓	✓	90.92	38.51
	✓	✓		✓	90.89	38.57
		✓	✓	✓	90.87	38.57

Table 4. Top-5 combinations of the **Tr**ack, **Re**lease, **Ar**tist, **La**bel and **St**yle Tags features in *MTAT*. The results are sorted according to the ROC-AUC values.

5. CONCLUSIONS

In this paper, we studied the usage of editorial metadata as a source of supervision for contrastive feature extraction models intended for music classification. We contributed to the previous work on metadata-based feature learning by scaling up the experiments in terms of dataset size, considering additional editorial metadata notions, and by experimenting with a new contrastive learning setup. We could validate that some of the observations from previous studies still hold for models trained on 10 times more data. Namely, we observed that some metadata-based representations are superior to their tag-based counterparts and that artist associations provide the best representations. Additionally, we found that the features learned from different metadata notions can be combined or jointly learned, showing slight performance improvements for particular tasks. To the best of our knowledge, this is the first study using Discogs' metadata to train music feature models. We did this by generating pairs of tracks according to metadata notions with simple matching rules, which leaves room for experimentation with the dataset.

In future research, we will explore more complex relationships in the metadata. For example, we could rely on metadata co-occurrences (e.g., record labels sharing many artists) to create more detailed associations. Another direction is to combine our approach with more sophisticated contrastive learning paradigms, which opens up a possibility for performance improvements. Finally, we will consider extending the evaluation to additional tasks such as music recommendation or arousal and valence regression.

6. ACKNOWLEDGEMENTS

This research was carried out under the project Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033, funded by the Spanish Ministerio de Ciencia e Innovación and the Agencia Estatal de Investigación.

7. REFERENCES

[1] A. van den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

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

[3] J. Lee, J. Park, K. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, 2018.

[4] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

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

[6] 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.

[7] 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," in *Proceedings of the ACM Web Conference 2022*, 2022.

[8] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, "S3t: Self-supervised pre-training with swin transformer for music classification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[9] 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.

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

[11] J. Park, J. Lee, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[12] J. Kim, M. Won, X. Serra, and C. C. S. Liem, "Transfer learning of artist group factors to musical genre classification," *Companion Proceedings of the Web Conference 2018*, 2018.

[13] J. Lee, J. Park, and J. Nam, "Representation learning of music using artist, album, and track information," in *International Conference on Machine Learning (ICML) 2019, Machine Learning for Music Discovery Workshop*, 2019.

[14] J. Kim, J. Urbano, C. C. S. Liem, and A. Hanjalic, "One deep music representation to rule them all? a comparative analysis of different representation learning strategies," *Neural Computing and Applications*, 2020.

[15] Q. Huang, A. Jansen, L. Zhang, D. P. Ellis, R. A. Saurous, and J. Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[16] A. Ferraro, X. Favory, K. Drossos, Y. Kim, and D. Bogdanov, "Enriched music representations with multiple cross-modal contrastive learning," *IEEE Signal Processing Letters*, 2021.

[17] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Learning music audio representations via weak language supervision," *arXiv:2112.04214v1 [cs.SD]*, 2021.

[18] A. Joorabchi and A. E. Mahdi, "An unsupervised approach to automatic classification of scientific literature utilizing bibliographic metadata," *Journal of Information Science*, 2011.

[19] J. K. Leung, I. Griva, and W. G. Kennedy, "Making use of affective features from media content metadata for better movie recommendation making," *arXiv preprint arXiv:2007.00636*, 2020.

[20] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020.

[21] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, no. Jan, 2003.

[23] D. Bogdanov and P. Herrera, "Taking advantage of editorial metadata to recommend music," in *International Symposium on Computer Music Modeling and Retrieval (CMMR'12)*, 2012.

[24] D. Bogdanov and X. Serra, "Quantifying music trends and facts using editorial metadata from the Discogs database," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[25] J. Burke, R. Rygaard, and Z. Yellin-Flaherty, "Clam!: Inferring genres in the discogs collaboration network," 2014.

[26] P. Budner and J. Grahl, "Collaboration networks in the music industry," *arXiv preprint arXiv:1611.00377*, 2016.

[27] N. Andrade and F. Figueiredo, "Exploring the latent structure of collaborations in music recordings: A case study in jazz." in *ISMIR*, 2016.

[28] L. Gienapp, C. Kruckenberg, and M. Burghardt, "Topological properties of music collaboration networks: The case of jazz and hip hop." *DHQ: Digital Humanities Quarterly*, 2021.

[29] D. Bogdanov, A. Porter, H. Schreiber, J. Urbano, and S. Oramas, "The acousticbrainz genre dataset: Multi-source, multi-level, multi-label, and large-scale," in *Proceedings of the 20th International Society for Music Information Retrieval (ISMIR)*, 2019.

[30] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Audio Based Disambiguation Of Music Genre Tags," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[31] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "COALA: co-aligned autoencoders for learning semantically enriched audio representations," in *Workshop on Self-supervised learning in Audio and Speech, International Conference on Machine Learning (ICML)*, 2020.

[32] ——, "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)*, 2021.

[33] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, 2020.

[34] 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.

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[36] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[37] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[38] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *Late-Braeking/Demo, International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[39] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "ESSENTIA: an audio analysis library for music information retrieval," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[40] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[41] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," *arXiv:1612.01840v3 [cs.SD]*, 2016.

[42] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, (ISMIR)*, 2009.

[43] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-Jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML)*, 2019.

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

[45] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," in *Proceedings of 17th Sound and Music Computing Conference (SMC)*, 2020.

[46] P. Tovstogan, D. Bogdanov, and A. Porter, "Mediaeval 2021: Emotion and theme recognition in music using jamendo," 2021.

[47] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

MELODY INFILLING WITH USER-PROVIDED STRUCTURAL CONTEXT

Chih-Pin Tan[1,2] **Alvin W.Y. Su**[1] **Yi-Hsuan Yang**[2,3]

[1] National Cheng Kung University, [2] Academia Sinica, [3] Taiwan AI Labs

p76091551@gs.ncku.edu.tw, alvinsu@mail.ncku.edu.tw, yang@citi.sinica.edu.tw

ABSTRACT

This paper proposes a novel Transformer-based model for music score infilling, to generate a music passage that fills in the gap between given past and future contexts. While existing infilling approaches can generate a passage that connects smoothly locally with the given contexts, they do not take into account the musical form or structure of the music and may therefore generate overly smooth results. To address this issue, we propose a structure-aware conditioning approach that employs a novel attention-selecting module to supply user-provided structure-related information to the Transformer for infilling. With both objective and subjective evaluations, we show that the proposed model can harness the structural information effectively and generate melodies in the style of pop of higher quality than the two existing structure-agnostic infilling models.

1. INTRODUCTION

In recent years, machine learning techniques have been widely applied to symbolic music generation. A large number of models attain *sequential generation* by accounting for only the *past context*, i.e., the generated music depends on only the preceding musical content [1–14]. While sequential generation can find useful use cases, it does not align with typical human compositional practices which can be non-sequential in nature. Musicians often write motifs or small pieces to get inspiration first, before working on the middle parts to connect them.

Hence, we focus on the scenario when both the past and *future* contexts are given, which is called *music score infilling* or inpainting [15]. As shown in Figure 1(a), the task is to let models fill in the missing part between the two given segments. Prompt-based conditioning approaches [15–23] have been applied to such a task in recent years, treating the two given segments as the "prompt." Among them, the variable-length infilling model (VLI) [20] obtains promising results by adding special positional encodings to XLNet [24], a permutation-based language model that is naturally suitable for generative tasks with given bi-

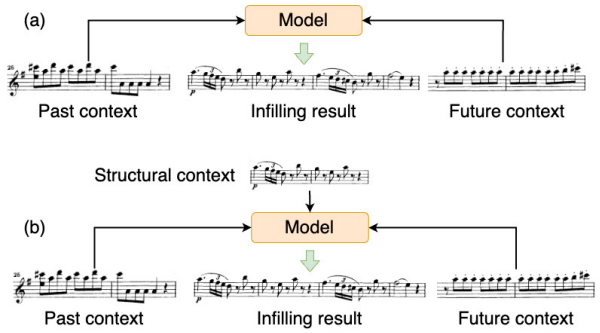

Figure 1: Comparison between (a) structure-agnostic and (b) structure-aware approaches for music score infilling.

directional contexts. The experiment of VLI shows that their model is capable of connecting the past and future contexts smoothly locally for infilling solo piano passages of up to 4 bars (measures).

Considering composers usually write musical pieces in a hierarchical manner [25], we note that prompt-based conditioning approaches have a strong limitation: they generate results with only consideration of local smoothness among the past context, future context, and result, without taking care of the overall musical *form* or *structure* of the music. For instance, a composer may like to write a song in a musical form of `ABA′B′`. If we consider the concatenation of the segments corresponding to `A` and `B` (i.e., `AB`) as the past context, and the segment corresponding to `B′` as the future context, and feed them to an existing infilling model, the model may generate a sequence that consists of similar melody and chord progression as the segments corresponding to `B` and `B′`, not the intended repetition or variation of the segment corresponding to `A`.

To address this issue, we propose in this paper a novel **structure-aware** setting for music infilling. As shown in Figure 1(b), besides the past and future contexts exploited by conventional structure-agnostic, prompt-based models, out approach additionally capitalizes for the infilling task the *structural context*, a music segment corresponding to a certain part of the whole music that is supposed to share the same *structure label* (such as `A` or `B`) with the missing segment. Accordingly, besides local smoothness, the model also needs to consider the similarity between the infilled segment and the structural context. Here, we assume the structural context is provided by a user, not generated by a model. For example, the user may designate the segment corresponding to `A` as the structural context, thereby

inform the model with the intended musical form.

We improve upon the VLI model [20] in the following ways to realize structure-aware infilling. First, we use the classic Transformer [26–28] instead of the more sophisticated XLNet [24] as the model backbone, to make it easier to add a conditioning module to exploit the structural context. To improve the capability of the Transformer to account for bi-directional contexts, we propose two novel components, the *bar-count-down technique* (Section 3.2) and *order embeddings* (Section 3.3), which respectively give the model an explicit control of the length of the generated music, and a convenient way to attend to the future context. Second, being inspired by the Theme Transformer [29], we use not a Transformer decoder-only architecture but a sequence-to-sequence (seq2seq) Transformer encoder/decoder architecture, using the cross-attention between the encoder and decoder as the conditioning module to account for the structural context. Moreover, we propose an *attention-selecting module* that allows the Transformer to access multiple structural contexts while infilling different parts of a music piece, which can be useful both in the training and inference time (Section 3.4) .

For evaluation, we compare our model with two strong baselines, the VLI [20] and the work of Hsu & Chang [21], on the task of symbolic-domain melody infilling of 4-bar content using the POP909 dataset [30] and the associated structural labels from Dai *et al.* [31]. With objective and subjective analyses, we show that our model greatly outperforms the baselines in the structure completeness of the generated pieces, without degrading local smoothness.

We set up a webpage for demos [1] and open source our code at a public GitHub repository. [2]

2. RELATED WORK

Generating missing parts with given surrounding contexts has been attempted by early works. DeepBach [17] predicts missing notes based on the notes around them. They use two recurrent neural networks (RNNs) to capture the past and future contexts, and a feedforward neural network to capture the current context from notes with the same temporal position as the target note. COCONET [16] trains a convolutional neural network (CNN) to complete partial musical scores and explores the use of blocked Gibbs sampling as an analog to rewriting. They encode the music data with the piano roll representation and treat that as a fixed-size image, so the model can only perform fixed-length music infilling. Inpainting Net [15] uses an RNN to integrate the temporal information from a variational auto-encoder (VAE) [32] for bar-wise generation, Wei *et al.* [23] build the model with a similar concept as Inpainting Net and use the contrastive loss [33, 34] for training to improve the infilling quality. Some Transformer-based models have also been proposed to achieve music infilling. Ippolito *et al.* [18] concatenate the past and future context with a special separator token. They keep the original positional encoding of the contexts and the missing segment, which again limits the length of given contexts and generated sequence to be fixed. We see that these infilling models impose some data assumptions and thereby have certain restrictions, e.g., the length of the input sequence cannot be arbitrary, or the missing segment needs to be complete bars. The work of Hsu & Chang [21] is free of these restrictions. They use two Transformer encoders to capture the past and future context respectively and generate results with a Transformer decoder. The VLI model [20] can also realize variable-length infilling. However, to our best knowledge, no existing models have explicitly considered structure-related information for infilling.

Structure-based conditioning has been explored only recently by Shi *et al*. [29] in their Theme Transformer model for sequential music generation. They use a seq2seq Transformer to account for not only the past context but also an additional pre-given theme segment that is supposed to manifest itself multiple times in the model's generation result. The present work can be considered as an extension of their work to the scenario of music infilling.

3. METHODOLOGY

Given a past context C_{past} and a future context C_{future}, the general, **structure-agnostic music infilling** task entails generating an *infilled segment* T that interconnects C_{past} and C_{future} smoothly, preferably in a musically meaningful way. When using an autoregressive generative model such as the Transformer as the model backbone, the training object is to maximize the following likelihood function:

$$\prod_{0<k\leq|T|} P(t_k|t_{<k}, C_{\text{past}}, C_{\text{future}}) \,, \tag{1}$$

where t_k denotes the element of T at timestep k, $t_{<k}$ the subsequence consisting of all the previously generated elements, and $|\cdot|$ the length of a sequence.

Extending from Eq. (1), we propose and study in this paper a special case, called **structure-aware music infilling**, where an additional segment G representing the *structural context* is given, leading to the new objective:

$$\prod_{0<k\leq|T|} P(t_k|t_{<k}, C_{\text{past}}, C_{\text{future}}; G) \,. \tag{2}$$

As depicted in Figure 2(a), our model is based on Transformer with the encoder-decoder architecture. It uses the decoder to self-attend to the prompt (i.e., C_{past} and C_{future}) and the previously-generated elements (i.e., $t_{<k}$), and the encoder to cross-attend to the structural context G. We provide details of the proposed model below.

Note that we do not require the length of all the involved segments to be fixed; namely $|T|$, $|C_{\text{past}}|$, $|C_{\text{future}}|$ and $|G|$ are all variables in our setting.

3.1 REMI-based Token Representation

To incorporate structure-related information to our representation of the music data, we devise an extension of the REMI-based representation [8] that comprises five types

[1] https://tanchihpin0517.github.io/structure-aware_infilling
[2] https://github.com/tanchihpin0517/structure-aware_infilling