Musical Duet Generation with Tree-Structured Variational Autoencoder

Adam Oudad[1,2] and Taketo Akama[2] and Hiroaki Saito[1]

[1] Keio University, Graduate School of Science and Technology, Department of Computer Science
adam.oudad@keio.jp
[2] Sony Computer Science Laboratories, Inc., Tokyo, Japan

Abstract. Recent successful latent space models based on Variational AutoEncoder (VAE) can generate polyphonic music for solo instrument. Polyphony with multiple tracks is more challenging in many aspects. Towards this goal, we propose the intermediate task of musical duet generation. In this setting, it is common to have to deal with two different instruments and note overlaps among tracks occur frequently. Unfortunately, these meaningful overlaps are discarded by current musical models. This limitation hinders their ability to generate multi-track music. We thus propose two data structures, *MergedTree* and *HierarchicalTree,* to overcome this limitation and three models, *MergedTree VAE*, *SharedHierarchicalTree VAE* and *HierarchicalTree VAE,* which leverage these data representations to reconstruct musical duets[3]. We evaluate our models on Lakh MIDI dataset and compare them to a *PianoTree VAE* baseline. Our *MergedTree VAE* model outperforms the baseline model on reconstruction scores. In addition, all proposed models are able to incorporate a statistically relevant ratio of overlaps among tracks.

Keywords: deep learning, algorithmic music composition, variational inference, data representation

1 Introduction

Automatic music composition is a challenging problem in music information processing. Deep learning techniques have significantly improved the ability of machines to generate more natural music (Ferreira & Whitehead, 2019; Wu & Yang, 2020; Wu & Yang, 2020; Curtis Hawthorne & Eck, 2018; Huang et al., 2018). Variational AutoEncoder (VAE) is an especially promising class for controllable music generation, yet the polyphonic and multitrack generation is still non-trivial (Wang, Zhang, et al., 2020). We base our research on a recent promising polyphonic solo generation model by Wang, Zhang, et al. (2020), and develop a polyphonic and multitrack generation approach. In this paper, we

[3] A demo of reconstructions and interpolations is available at https://adamoudad.github.io/aimc2021/.

focus on an intermediate task toward this goal: generating musical duets, and leave generation with more tracks for future work.

We define a **musical duet** as music for two instruments. A musical duet differs in this regard from a musical solo on two main properties. Firstly, two instrument tracks should be playing at the same time, and secondly, overlaps between notes are allowed and may occur. The second property makes it different from chorale generation for example, in which overlaps are usually forbidden. In our setting, we refer to the two tracks of a musical duet as *melody* and *accompaniment*. We define an **overlapping** among instruments as the occurence of notes with the same pitch being played simultaneously by at least two instruments. Notes do not need to share the same onset timing, and may overlap on several timesteps in which case we would count each timestep as different overlaps.

We propose two data structures, *MergedTree* and *HierarchicalTree*, for representing a musical duet, which can be trivially extended to more tracks. We train three different architectures, *MergedTree VAE*, *HierarchicalTree VAE* and *SharedHierarchical VAE* leveraging these two data representations in the musical duet setting on Lakh MIDI dataset, in addition to a *PianoTree VAE* baseline. We demonstrate the statistical relevance of overlaps and the variety of instruments assigned to either melody or accompaniment tracks in the dataset. Finally, we propose an objective evaluation of generative models' ability to reproduce overlaps between tracks.

2 Related works

Multi-track generation has been addressed by Dong, Hsiao, Yang, and Yang (2018) using generative adversarial networks, and by Simon et al. (2018) with variational autoencoder. In addition to their generative power, variational autoencoders can be used to control generated music (Akama, 2020; Akama, 2019; Pati, Lerch, & Hadjeres, 2019; Roberts, Engel, Raffel, Hawthorne, & Eck, 2018; Wang, Zhang, et al., 2020; Brunner, Konrad, Wang, & Wattenhofer, 2018; Dong et al., 2018; Tan & Herremans, 2020) and to create smooth musical transitions by interpolating the variational latent space. A common representation for musical data is a MIDI-like representation and has been used by Simon et al. (2018) to generate multi-track music, yet this representation often leads to unsatisfying generation (Dong et al., 2018; Yang, Chou, & Yang, 2017). On the other part, the *PianoTree* representation proposed by Wang, Zhang, et al. (2020) attempts to address this by using a representation based on the pianoroll representation, but is unable to represent multi-track scores.

3 Method

Our three proposed models use a variational autoencoder architecture to model music (Kingma & Welling, 2014). This architecture is explained in section 3.2. The differences between models come from the underlying data structure they operate on, which is detailed in section 3.1 and Fig. 1.

3.1 Data structures for representing multi-track music

The *PianoTree* data structure proposed by Wang, Zhang, et al. (2020) represents a musical pianoroll in a tree structure. Leaf nodes of this structure are individual notes arranged by pitch in ascending order. We extend this representation to overcome its limitations in the case of multiple tracks. Each track pianoroll is viewed as a separate *PianoTree* structure. Tracks are combined in two different ways.

The *HierarchicalTree* combines the two *PianoTrees* by adding a new hierarchical layer for tracks. The root node is connected to nodes each corresponding to one track, which is a *PianoTree* representation of this track's pianoroll.

The *MergedTree* merges the two *PianoTrees* by first shifting the pitch ranges of tracks to avoid pitch overlap. In our duet setting, with two tracks, we shift the accompaniment note pitches by 131, so that a melody pitch ranges between 0 and 127 values while an accompaniment pitch ranges between 131 and 258 values. Likewise in *PianoTree* data structure, pitch values 128, 129 and 130 are reserved for start of sequence (SOS), end of sequence (EOS) and padding (PAD) tokens. We then merge the melody *PianoTree* with the accompaniment *PianoTree* by appending at each timestep node of the melody *PianoTree* the leaf nodes of the corresponding timestep node in the accompaniment *PianoTree*. We obtain a data structure as described by Fig. 1. In our code implementation, all remaining slots for notes at a given timestep are filled with PAD tokens.

These two data structures can be trivially extended to more tracks by repeating the process described above which adds new tracks.

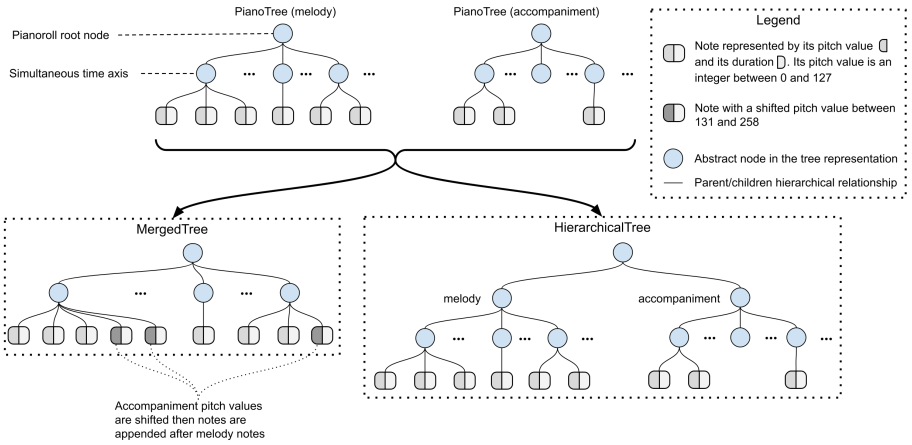

Fig. 1: Tree data structures for musical duet representation.

3.2 Variational autoencoder

The variational autoencoder (*VAE*) consists of an encoder $q_\phi(z|x)$ that approximates the posterior distribution of the latent variable $p(z|x)$ and a decoder $p_\theta(x|z)$ that models the reconstruction from a prior normal distribution $p(z)$.

The following objective function is optimized by the model during training.

$$\mathcal{L}(\phi, \theta; x) = -\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x|z) + \beta KL(q_\phi(z|x)||p(z))$$

The first term is the *reconstruction term* and controls the quality of reconstructions. The second term controls the shape of the latent distribution so that it matches that of a normal distribution. This constraint ensures sampled latent vectors in-between embeddings of the ground-truth data produce plausible generated outputs.

Our implementation of a *VAE* for music generation follows the *PianoTree VAE* proposed by Wang, Zhang, et al. (2020). We embed each note into a 64-sized vector. The *PianoTree* data structure is then embedded using a simultaneous note axis (see Fig. 1) bidirectional gated recurrent unit (GRU, (Chung, Gülçehre, Cho, & Bengio, 2014)) of hidden size 128. Each timestep is then embedded using a time-axis (or vertical axis as in Fig. 1) bidirectional GRU of hidden size 256. We obtain a fixed representation of the input *PianoTree* which is linearly mapped to the mean and variance parameters of a multivariate gaussian distribution of 256 variables. Decoding starts by sampling a latent code using this distribution. Symmetrically to the encoder, a time-axis bidirectional GRU decodes each 512-sized timestep vectors, followed by a simultaneous note axis GRU which decodes notes as 256-sized vectors. Finally, a note's pitch value is decoded with a linear map and duration is decoded to a multihot vector using a GRU.

3.3 Proposed models

Fig. 2 shows a graphical representation of all models detailed in this section.

Building on the two new data structures detailed in section 3.1, we propose the three following models.

- The *HierarchicalTree VAE* assigns a separate autoencoder to each track of the *HierarchicalTree*. We first embed the notes of the input musical segment with a different embedding layer for each track. This embedded *HierarchicalTree* structure is then fed to the corresponding encoder for each track. The outputs of the track-specific encoders are combined with a conductor encoder to obtain a latent code. This embedding vector of the whole musical segment is decoded by a conductor decoder which assigns to each track decoder one embedding vector. Each of these track-specific embedding vectors is decoded using the corresponding track decoder.
- The *SharedHierarchicalTree VAE* has the same architecture as the *HierarchicalTree VAE* except that the track autoencoders weights are shared, resulting in an architecture similar to the proposed hierarchical architecture of the MusicVAE extended for multi-track by Simon et al. (2018). Similarly

to the *HierarchicalTree VAE*, note embeddings fed to the shared autoencoder are learned separately for each track.

– The *MergedTree VAE* is a *PianoTree VAE* operating on *MergedTree* data structure instead of the *PianoTree* data structure. Its embedding layer is extended to embed additional shifted pitch values of the accompaniment track.

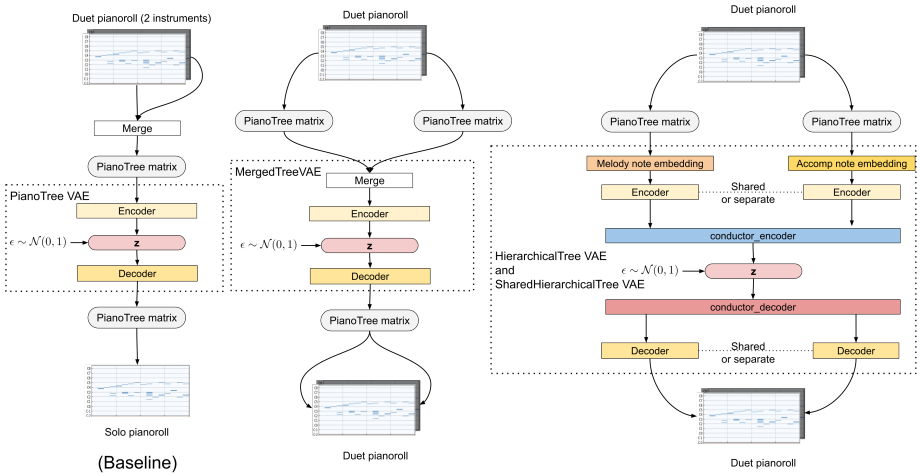

Fig. 2: PianoTree VAE baseline architecture compared with MergedTree VAE, HierarchicalTree VAE proposed models. SharedHierarchicalTree VAE shares the same architecture as HierarchicalTree VAE, but the track encoders and decoders share weights.

4 Experiments

4.1 Dataset

The Lakh MIDI dataset is a collection of 178,561 MIDI files scraped from the web (Raffel, 2016). The dataset contains multi-instrument music from a wide range of musical genres. We extract melody and accompaniment tracks using *midi-miner* library (Guo, Simpson, Magnusson, Kiefer, & Herremans, 2020). This step discards all MIDI files in which anyone of melody and accompaniment track is missing. This share represents 62,948 files, which is 35% of all the files in the dataset. Each remaining MIDI file is quantized with a beat resolution of 4, which corresponds to a quantization step of a 16th note. We convert MIDI files to pianorolls using pypianoroll (Hao-Wen Dong & Yang, 2018). We restrict the note pitch range between 30 and 100. Following Wang, Zhang, et al., 2020 we limit the

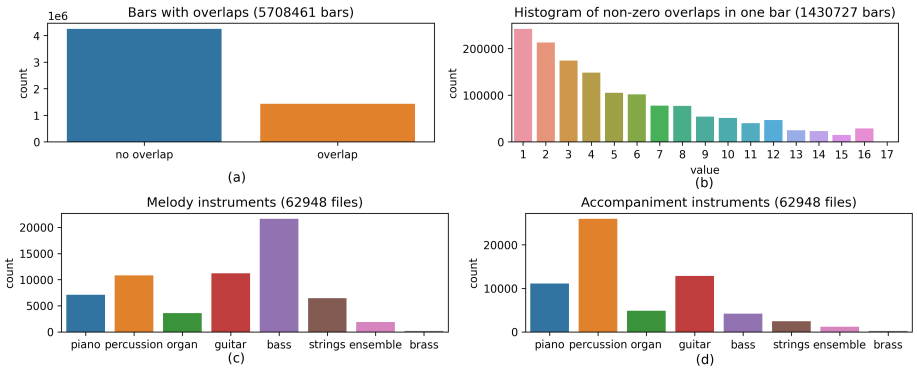

Fig. 3: Statistics computed on Lakh MIDI dataset. (a) shows the number of bars with overlaps compared to the number of bars without overlaps. (b) shows a histogram of bars with non-zero overlaps, with the number of overlaps in one bar in horizontal axis and the counted number of bars on vertical axis. (c) and (d) show the histograms of instrument categories assigned to melody and accompaniment. The instrument category is inferred using the MIDI instrument program. The MIDI standard defines 16 different instrument categories which are piano, percussion, organ, guitar, bass, strings, ensemble, brass, reed, pipe, synth led, synth pad, synth effect, ethnic, percussive and sound effects.

number of simultaneous notes in a data sample to 16, discarding samples that do not satisfy these conditions for both melody and accompaniment. We finally slice each pianoroll into samples of 16 timesteps for one bar and 32 timesteps for two bars music segments.

In Fig 3, (a) and (b) demonstrate the statistical relevance in the data of what we refer to as *multi-track overlapping problem*. From the 5,684,745 pairs of melody and accompaniment bars, 1,430,727 bars contain overlapping notes, which is 25.17% as shown on the histogram (a) of Fig 3. The histogram (b) in Fig. 3 shows a finer view of the number of overlaps found in each bar of the dataset. We find a frequency of 1.334 overlaps per bar, that is 2.668 overlaps per 2-bar segments in the dataset.

We further study the differences between melody and accompaniment tracks by counting the different instruments associated with melody or accompaniment. We consider the instrument categories defined in the general MIDI standard, which are piano, percussion, organ, guitar, bass, strings, ensemble, brass, reed, pipe, synth lead, synth pad, synth effect, ethnic, percussive, and sound effects. We find that 80.1% of the MIDI files in the data associate different instrument categories to melody and accompaniment. In Fig. 3, (c) and (d) show these associations. In particular, we note that the preferred category is *bass* for melody and *chromatic percussion* (which contains instruments such as vibraphone and celesta) for accompaniment. This counter-intuitive predominance of bass cate-

gory for melody tracks suggests that bass lines are often similar to melody lines and tend to be confused by the *midi-miner* classifier.

4.2 Training and evaluation

Table 1: Model evaluation on the reconstruction of 1-bar (top) and 2-bars (bottom) pianorolls after 300k steps of training. Baseline is *PianoTree VAE*. HT, SHT, and MT respectively denote *HierarchicalTree VAE*, *SharedHierarchicalTree VAE* and *MergedTree VAE*. SHTf refers to *SharedHierarchicalTree VAE* with flat conductors and SHTg refers to the same model with GRU conductors.

1-bar models	Baseline	Ours[HT]	Ours[SHTf]	Ours[SHTg]	Ours[MT]
Onset precision	0.898	0.681	0.779	0.760	**0.921**
Onset recall	0.858	0.842	0.451	0.844	**0.910**
Onset F1	0.878	0.753	0.571	0.799	**0.915**
Duration precision	0.956	0.950	0.890	0.961	**0.967**
Duration recall	0.971	0.906	0.867	0.958	**0.972**
Duration F1	0.964	0.928	0.878	0.960	**0.969**
Overlapping ratio	0(1.334)	1.716(0.382)	16.787(15.453)	1.611(0.277)	**1.599(0.225)**

2-bars models	Baseline	Ours[HT]	Ours[SHTf]	Ours[SHTg]	Ours[MT]
Onset precision	0.810	0.214	0.355	0.344	**0.851**
Onset recall	0.706	0.400	0.277	0.507	**0.774**
Onset F1	0.755	0.278	0.311	0.410	**0.810**
Duration precision	0.911	0.799	0.822	0.791	**0.934**
Duration recall	**0.942**	0.574	0.754	0.745	0.939
Duration F1	0.926	0.668	0.786	0.767	**0.937**
Overlapping ratio	0(2.668)	1.950(0.718)	42.963(40.295)	3.096(0.428)	**2.592(0.076)**
Rel. training speed	**1**	1.807	1.564	1.728	1.107

We transpose note pitches in each pianoroll by 5 semitones down up to 6 semitones up to augment the dataset. We train all models using Adam optimizer (Kingma & Ba, 2015) with teacher forcing, and we set the learning rate to $5e^{-4}$. We anneal the β term in front of Kullback-Leibler divergence from 0 to 0.1 on the first 200,000 training iterations. All models converge after 300,000 training iterations. Evaluation is computed on 500 batches of 32 data samples each. Table 1 reports evaluation results for all models on 1-bar and 2-bars pianorolls. The relative training speed of each model is reported by dividing the total training time of a model by the total training time of the baseline. Onset and duration metrics are the metrics defined in Wang, Zhang, et al. (2020) and are calculated on the correctly reconstructed note pitch values and the note duration multihot vectors. We believe the difference we see in our results compared to *PianoTree*

VAE baseline in Wang, Zhang, et al. (2020) comes from the different nature of datasets used. We use Lakh MIDI dataset which is a more diverse MIDI collection than the dataset used in Wang, Zhang, et al. (2020). Their dataset consists of mostly solo instrumental music from POP909 dataset (Wang, Chen, et al., 2020) and Musicalion website. On the contrary, the data we use has much more instrumental variety as demonstrated in section 4.1.

We define the overlapping ratio as the mean number of overlaps between tracks reconstructed by the models. Our goal is to reproduce the overlap calculated in the dataset in section 4.1 by reconstructing music with an overlapping ratio sensibly close to the ground-truth, which are 1.334 overlaps for 1-bar and 2.668 overlaps for 2-bar segments. We report the absolute difference between the overlapping ratio calculated on the reconstruction to these two ground-truth values. Because the PianoTree VAE baseline has no capability of separating melody and accompaniment parts, its overlapping ratio is set to 0.

MergedTree VAE consistently outperforms all other models on onset and duration scores and is able to reconstruct music with the closest overlapping ratio to ground-truth. This reveals its suitability for overcoming the challenges in multi-track music. *SharedHierarchicalTree VAE* with GRU conductors is the overall best performing model using *HierarchicalTree* representation. Surprisingly, all *HierarchicalTree* based models see their performances drop for 2-bar pianorolls compared to 1-bar pianorolls. The results strongly favor a timestep-level track hierarchy as described by *MergedTree* representation, to a bar-level track hierarchy used in *HierarchicalTree* based models.

We provide interpolation and generation samples on the demo website at `https://adamoudad.github.io/aimc2021/`.

5 Conclusion

We presented three models for generating musical duets *MergedTree VAE*, *HierarchicalTree VAE* and *SharedHierarchicalTree VAE*. They are built on top of two data structures which extend *PianoTree* representation to multi-track music. Our proposed *MergedTree VAE* model outperforms the baseline on all scores calculated on the Lakh MIDI dataset. Our models can be used to separate tracks of a pianoroll and generate richer multi-track music with natural overlaps. We hope that our work opens up new directions for developing more effective data structures and useful models for polyphonic and multitrack music generation.

References

Akama, T. (2019, November). Controlling Symbolic Music Generation based on Concept Learning from Domain Knowledge. In *Proceedings of the 20th International Society for Music Information Retrieval Conference* (p. 816-823). Delft, The Netherlands: ISMIR. doi: 10.5281/zenodo.3527936

Akama, T. (2020, October). Connective fusion: Learning transformational joining of sequences with application to melody creation. In *Proceedings of*

the 21st International Society for Music Information Retrieval Conference (p. 46-53). Montreal, Canada: ISMIR. doi: 10.5281/zenodo.4245360

Brunner, G., Konrad, A., Wang, Y., & Wattenhofer, R. (2018). MIDI-VAE: modeling dynamics and instrumentation of music with applications to style transfer. In E. Gómez, X. Hu, E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018* (pp. 747–754).

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR, abs/1412.3555*.

Curtis Hawthorne, D. I., Cheng-Zhi Anna Huang, & Eck, D. (2018). Transformer-nade for piano performances. *submission, NIPS Second Workshop on Machine Learning for Creativity and Design*.

Dong, H., Hsiao, W., Yang, L., & Yang, Y. (2018). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (pp. 34–41). AAAI Press.

Ferreira, L., & Whitehead, J. (2019). Learning to generate music with sentiment. In A. Flexer, G. Peeters, J. Urbano, & A. Volk (Eds.), *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019* (pp. 384–390).

Guo, R., Simpson, I., Magnusson, T., Kiefer, C., & Herremans, D. (2020). A variational autoencoder for music generation controlled by tonal tension. In *Joint Conference on AI Music Creativity (CSMC + MuMe)*.

Hao-Wen Dong, W.-Y. H., & Yang, Y.-H. (2018). Pypianoroll: Open source python package for handling multitrack pianorolls. In *Late-Breaking Demos of the 19th International Society for Music Information Retrieval Conference (ISMIR)*.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., ... Eck, D. (2018). Music transformer: Generating music with long-term structure. *arXiv preprint arXiv:1809.04281*.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *CoRR, abs/1312.6114*.

Pati, A., Lerch, A., & Hadjeres, G. (2019). Learning to traverse latent spaces for musical score inpaintning. In *20th International Society for Music Information Retrieval Conference (ISMIR)*. Delft, The Netherlands.

Raffel, C. (2016). *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching* (Unpublished doctoral dissertation). Columbia University.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (Vol. 80, pp. 4361–4370). JMLR.org.

Simon, I., Roberts, A., Raffel, C., Engel, J., Hawthorne, C., & Eck, D. (2018). Learning a latent space of multitrack measures. *Machine Learning for Creativity and Design, NeurIPS 2018 Workshop*, *abs/1806.00195*.

Tan, H. H., & Herremans, D. (2020). Music fadernets: Controllable music generation based on high-level features via low-level feature modelling. In *Proc. of the International Society for Music Information Retrieval Conference.*

Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., ... Xia, G. (2020). POP909: A pop-song dataset for music arrangement generation. *CoRR*, *abs/2008.07142*.

Wang, Z., Zhang, Y., Zhang, Y., Jiang, J., Yang, R., Zhao, J., & Xia, G. (2020). Pianotree vae: Structured representation learning for polyphonic music. In *Proceedings of the 21st International Conference on Music Information Retrieval (ISMIR 2020).*

Wu, S., & Yang, Y. (2020). The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures. *CoRR*, *abs/2008.01307*.

Yang, L., Chou, S., & Yang, Y. (2017). Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In S. J. Cunningham, Z. Duan, X. Hu, & D. Turnbull (Eds.), *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017* (pp. 324–331).