Self-Attention mechanism, which allows it to model relationships between all elements of a sequence. The mechanism takes in matrices Q, K, and V, corresponding, respectively, to Queries, Keys and Values, and calculates a weighted sum of the values where the coefficients are given by a similarity function between the queries and the keys.

One of the models developed with the intent of decreasing the computational costs associated with the calculation of Attention matrices [17–20] is the Linear Transformer [18], which, as the name suggests, employs a version of the attention mechanism with computational and memory requirements that grow linearly with respect to sequence length (standard Attention has quadratic requirements), and is also faster than regular attention. This is achieved via the substitution of the standard softmax similarity score by one that allows the matrix multiplications necessary for the calculation of the attention matrix to be factorized in a more efficient way. If that kernel has a feature representation $\phi(x)$, one can write the Attention matrix as:

$$A(Q,K,V)_i = \frac{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j) V_j}{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)}. \quad (7)$$

With this expression, it is possible to calculate the factor inside the sum only once, and to reuse it to find all the queries. Specifically, the authors use the kernel with a feature map that results in a positive similarity function: $\phi(x) = \text{elu}(x){+}1$, where elu is the exponential linear function [21]. In this same work, the authors also discover an analogy between the Transformer and the RNN. We refer to [18] for further details on the Linear Transformer.

2.3 Generative Models of Music

As of the writing of this paper, most of the state-of-the art generative models of symbolic music are Transformers or Transformer-based. Music Transformer [22] uses relative self-attention [23] to process longer sequences of data and generate music that exhibits long-term structure. MuseNet [24] is able to produce multi-track compositions spanning several musical genres and artists by employing time, note and structural embeddings that give the model more context.

Midinet [25] and MuseGAN [26] are GAN-based generative models of music that use Convolutional Neural Networks (CNNs) as both the Generator and Discriminator. In those works, music is represented in the form of piano-rolls that work analogously to images. Most similar to our work are [27], where the Discriminator is trained to judge the content both locally and globally, and [13], which uses a Transformers-XL [28] as Generator and a pre-trained BERT [29] as Discriminator, and also employs the Gumbel-Softmax trick discussed above.

Several works have also explored conditioning musical generation with emotional content. In [30], human emotions are captured from images of human faces and then categorized into 7 categories. Then, the researchers try to generate music based on these emotions. In [31], Biaxial LSTM networks are used to produce polyphonic music, and the generation can be conditioned by emotion via 4 parameters originating from the valence and arousal dimensions. In [32], the authors translate between the visual and musical domains via emotions. Specifically, they use neural networks to extract emotional content from images and music, and subsequently feed the content originated from both pieces of data into a network to condition music generation. Music Fadernets [33] constitute a framework that can infer high-level feature representations by first modelling their equivalent low-level attributes, which are easier to quantify. These low-level features are then used for style transfer across arousal states. In [34], a model dubbed Bardo Composer is presented as a system to generate backgorund music for tabletop role-playing games. This is done through Stochastic Bi-Objective Beam Search (SBBS), a search algorithm that samples from a distribution of sequences and selects for one that maximizes for realism and emotion. Furthermore, [35] uses a genetic algorithm to influence specific LSTM units that learn to encode sentiment in a pre-training language modeling stage, allowing it to steer the passages that it generates towards a desired affective state. In [36], the authors use pre-defined mood-tags associated with each chord in a progression to guide the generative process step-by-step. The EMOPIA dataset [37] contains musical passages separated into four quadrants that each corresponds to a combination of positive or negative arousal and valence. The excerpts on the dataset are from piano transcriptions of pop songs that were labeled by its authors. In this same work, the authors also use a Transformer model that employs the Compound Word representation [38] to generate songs conditioned by affective states.

One of the factors that influence the final quality of the samples generated by models of symbolic music is the representation used to train them. For contemporary styles, like Pop or Hip-Hop, where a rigid metrical grid is often followed, it is desirable to incorporate data about the rhythmic structure of the songs into the representation. REMI [39], which stands for revamped MIDI-derived events, is a beat-based approach to modeling music that encodes this information through tokens that signal the beginning of a bar and the passage of each beat, while still maintaining some flexibility by allowing local tempo changes. More specifically, there NOTE_ON, NOTE_DURATION, VELOCITY, TEMPO, BAR and BEAT. A NOTE_ON event indicates the start of a note, NOTE_DURATION corresponds to the duration of that note, and VELOCITY is a parameter that indicates the intensity with which the keys on a piano are played, and correlates to the volume of the note produced. Finally, TEMPO dictates the tempo of the musical excerpt from the moment the token is produced forward, and BAR events indicate the start of a new bar. The Compound-Word Transformer [38] uses a separate embedding for each type of musical element (pitch, chord, tempo value, etc.).

3. METHODS

3.1 Architecture and Loss functions

Both the Generator and Discriminator are Transformers with linear versions of the Attention Mechanism [18]. Each of the models is composed by 6 Attention blocks.

The Generator has two roles. In the first place, it has to predict each item of each sequence from the real dataset based on the previous elements, that is, it has to complete pieces of already existing sequences. In standard Transformer fashion, a lower triangular or look-ahead mask is applied to the original sequence in order to prevent the model from seeing its future elements. The second objective of the network is to generate sequences that are similar to those of the real set from scratch, that is, without context from the real dataset, such that these sequences can fool the Discriminator. These sequences are generated step-by-step in an autoregressive manner, which is done via the Transformer-RNN analogy made in [18]. ¨

The Generator is conditioned via special scale and bias parameters that influence the Layer Normalization [40] layers existing within the Attention mechanism. There is one of these couples for each class on the dataset and one for the unlabeled sequences. Formally, if i, k and c stand, respectively, for position in the sequence, feature channel and class label, we have:

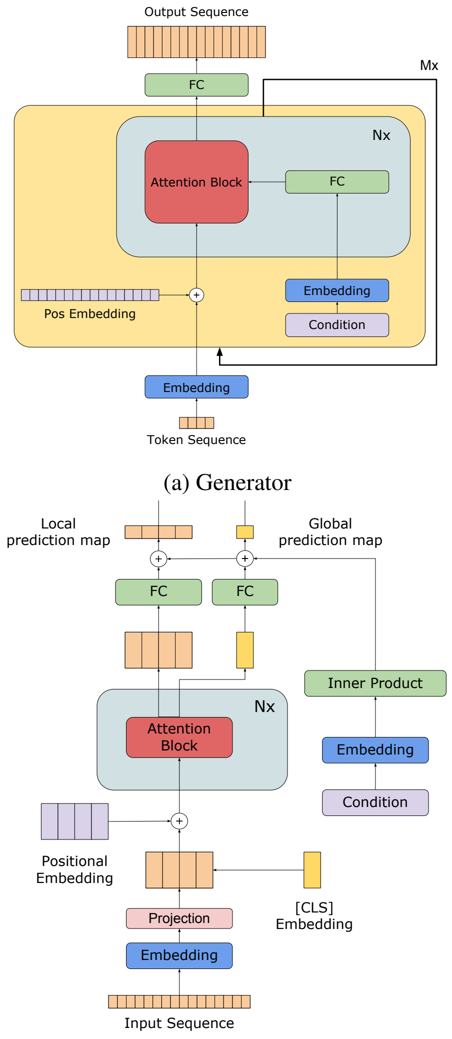
$$s'_{i,k} = \gamma_k^c \cdot s_{i,k} + \beta_k^c \tag{8}$$

where s and s' are input and output sequences, and γ and β are scale and bias.

The Discriminator takes each sequence as a whole and tries to determine if it is real or fake. To design this network, we took inspiration from the Visual Transformer [15], separating the sequences into patches of a certain length and transforming each patch into a single feature vector.

Our Discriminator has two outputs. First, there is a single feature unit that indicates if the passage originates from the real or fake datasets and if it exhibits the desired characteristics provided by the conditional signal. To produce this output, a [CLS] token is concatenated to the input sequence, similarly to BERT [29]. Then, the conditional information is incorporated via an inner product between an embedding of this information and the [CLS] representation. This essentially means that the model works as a Projection Discriminator [41]. The second output is a prediction map where each unit corresponds to a single patch in the sequence, that is, for every collection of musical symbols with length equal to the patch size, there is a value predicting whether that patch is real or fake. This technique, often used in image generating-GANs [42], ensures that the model prioritizes local structure.

Each of these two outputs serves the purpose of incorporating priors about musical structure into our model. A common way to frame the task of musical generation is as a language modelling one: each individual symbol is treated as a word, and these words compose phrases, periods, and so on. Using this analogy, the goal behind the proposed local loss is to inform if each particular sentence in a text is realistic or not, or, translating that to music, if every short musical idea in the form of a phrase or part of a phrase is realistic or not. The global prediction unit, on the other hand, acts as a signal that encapsulates the overall quality of the sequence and its ability to convey the desired emotional state, complementing the local prediction map. The networks are illustrated in Figure 1.



(a) Generator

(b) Discriminator

Figure 1: Generator and Discriminator. FC stands for Fully-Connected Layer. N is the number of self-attention blocks from which the models are made. M represents the number of autoregressive steps the Generator executes, that is, the sequence length.

3.2 Datasets

We used two datasets to train our models, mainly as a means to allow the networks to use a larger training corpus. Both consist only of songs performed on piano. The AILABS17k dataset [38] contains over 108 hours of piano covers of pop songs automatically transcribed by a state-of the art piano transcription model [43] and converted into MIDI files. The EMOPIA dataset [37] was constructed in a similar fashion to the one above, but its songs were afterwards labeled by the authors of the dataset according to perceived sentiment [2]. The clips on this dataset amount to approximately 11 hours. We used AILABS 17K to allow the networks to learn a general representation of music in a larger corpus, while also being trained on a smaller collection of songs that contains annotated data.

3.3 Training

We use the data representation proposed in [39], which consists of the NOTE_ON, NOTE_DURATION, VELOCITY, TEMPO, BAR and BEAT events described previously. Before the introduction of the Discriminator, the Generator was trained until convergence through the teacher forcing method (26000 steps). This pre-training stage guaranteed the stability of the adversarial stage that was to follow [13, 27, 44]. In this stage, the Generator was trained simultaneously on both datasets, and taking into consideration the difference in size between these datasets, to balance the training process, we alternated between optimization steps on randomly sampled batches from each set. The network worked with sequences of length 2048, corresponding, on average, to 1 minute of content.

For the adversarial stage, we used sequences of size 128. In order to reduce the effects of gradient variance that are inherent to the sampling process, sequences with length 16 originated from the real set were given to the Generator such that it could have a starting point to perform generation [27]. The Discriminator worked with subsequences of length 16, and the training was done exclusively on the EMOPIA dataset [37].

The networks were trained via a combination of the teacher forcing objective plus the RSGAN objective [8] with gradient penalty [9]. We performed 1 optimization step of the Discriminator per Generator step, using a learning rate of $1 \cdot 10^{-4}$. In total, 26000 global optimization steps were performed. The overall objective for the generator in this training stage is:

$$\mathcal{L}_G = \mathcal{L}_{\text{MLE}} + \alpha \mathcal{L}_{\text{RSGAN}_G\text{-global}} + \beta \mathcal{L}_{\text{RSGAN}_G\text{-local}}, \quad (9)$$

$$\text{where } \mathcal{L}_{mle} = -\mathbb{E}_{x \sim P}[\log G_\theta(x)] \quad (10)$$

where the factors $\mathcal{L}_{\text{RSGAN}_G\text{-global}}$ and $\mathcal{L}_{\text{RSGAN}_G\text{-local}}$ are respectively the global and local GAN losses detailed above, α and β, which we empirically chose to be equal to 1, are hyperparameters controlling the relative intensity of each loss factor, and $\mathcal{L}_{\text{MLE}}$ is the Maximum Likelihood. The Discriminator was simply trained with the local and global RSGAN objectives given previously in Equation 2 plus the global and local gradient penalties based on Equation 4 and regulated by a hyperparameter λ (which as per [9], we chose to be 10) . This loss is expressed as Equation 11. An algorithm outlining the training scheme is available in the supplementary material.

$$\begin{aligned} \mathcal{L}_D = &\mathcal{L}_{\text{RSGAN}_D\text{-global}} + \beta \mathcal{L}_{\text{RSGAN}_D\text{-local}} + \\ &\lambda(\mathcal{L}_{\text{GP-global}} + \mathcal{L}_{\text{GP-local}}). \end{aligned} \quad (11)$$

4. EXPERIMENTS

We evaluated our models both with respect to the overall quality of the samples they produce and their ability to generate songs that convey the conditioning emotional signals. To achieve this purpose, we used both the automatic evaluation metrics proposed in [26, 45] and a set of human evaluation metrics to compare our GAN with the system that, as far as we know, corresponds to a state-of-the-art generative model of symbolic music conditioned by sentiment currently available in the literature. Specifically, our model was compared with the Compound-Word Transformer [38] variation used by the authors of the EMOPIA article [37] to generate music conditioned by emotional class. We also compared our adversarially trained model with a Vanilla Transformer model that was not trained with the adversarial scheme. This model corresponds to the version of the generator network obtained after the pretraining was completed. Code for this work, along with audio samples, are available at [1] .

For the automatic metrics, we chose Pitch Range (distance between highest and lowest pitch), Number of Pitch Classes, and Polyphony (the average number of simultaneous notes). These metrics were calculated using the Muspy library [46]. For each model, we generated 400 samples (100 for each class) and evaluated these samples with respect to the characteristics above, then averaged the results to produce the overall model score. The results are presented in Table 1.

	PR	NPC	POLY
Real Data (EMOPIA) [37]	50.94	8.50	5.60
Baseline [37]	49.76	**8.52**	4.36
Transformer	48.79	8.65	4.37
Transformer GAN	**50.73**	9.45	**4.43**

Table 1: Comparison between the samples generated by ours and a state-of-art model. PR stands for Pitch Range, NPC is Number of Pitch Classes and POLY is Polyphony. The best results are highlighted in bold.

As it can be seen, our model obtains a superior performance than that of the baseline and the pre-trained model on two of the three metrics. Furthermore, it should be mentioned that the Generator is significantly smaller than the baseline, having only 6 Attention layers, while the latter has 12. To be more precise, the baseline has $\sim 40M$ parameters, while our generator has $\sim 25M$ and our Discriminator has $\sim 27M$. Also relevant is the fact that the representation used to train the baseline, that is, the Compound-Word representation [38], is more complex than the REMI representation that we use [39], and has also been proved to be better than REMI. Since the focus of our work was to implement the GAN framework within the context of symbolic music generation conditioned by sentiment, and given the complexity of this framework, especially when it is applied to the discrete domain, we chose to use a simpler representation in order to maintain the focus of our work on the implementation of the GAN. But as

[1] http://github.com/pneves1051/transformers_sentiment

our results show, adapting it to work within the adversarial context, which we leave to future work, could bring about even better results.

Finally, we performed a survey in which participants were asked to judge the musical excerpts generated by the models both in terms of their overall quality and their ability to convey the desired sentiments. Specifically, participants rated the samples on a 5-point Likert scale that ranged from very low to very high with respect to the following characteristics: Human-likeness, Originality, Structure, Overall Quality, Valence, and Arousal. Details from each characteristic and the corresponding scale are given in the supplementary material. The participants were recruited from the researchers' online circles. Each of the participants had to listen to 12 musical excerpts in total, 4 from each model, and within those 4 item groups, one from each of the 4 emotional classes. Before the test, some text explaining the basic concepts behind the research was shown to the participants. In total, 18 individuals participated in the experiment.

We present the average participants' scores given to each model for Human-likeness, Originality, Structure and Overall Quality in Table 2. These results suggest that both the proposed Transformer, and the Transformer GAN, are competitive with a state-of-the-art model with respect to the four qualitative metrics.

On the next step of the evaluations, we took the participants' answers to the questions related to Valence and Arousal and compared them to the real emotional labels provided to the model during the generation process. Figure 2 illustrates the results of this experiment.

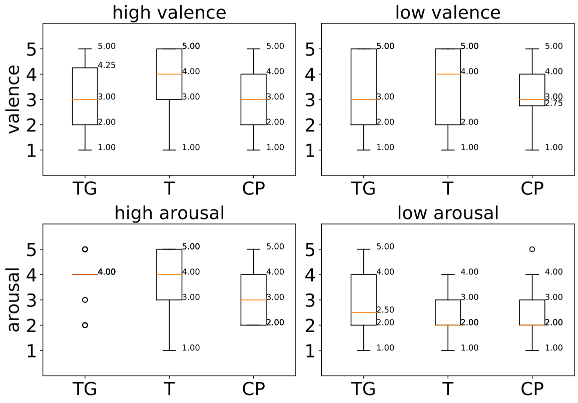


Figure 2: Results of the experiment where participants rated musical samples according to their perceptions about valence and arousal. The acronyms TG, T and CP correspond, respectively, to the Transformer GAN, Transformer and Compound-Word Transformer Baseline models.

Once again, we find that the models we developed obtain a performance that competes with that of the current state-of-the-art. By comparing the medians and quartiles of these boxplots, we can draw several conclusions. Firstly, all models seem to have more difficulty capturing valence than they do arousal. This may be an indication that musical aspects which have a great impact over arousal, such as velocity and tempo, are more easily understood by neural networks, while factors which have an impact over valence, such as modality or frequency of harmonic change [47], are more difficult to model for these systems.

Furthermore, we see that, in general, our models do not stand behind when compared to the baseline. Between the three, by observing the boxplots, it seems that while the Transformer and the Compound-Word baseline sometimes produce samples that situate themselves more strongly to the side to which they theoretically pertain (e.g., for the high valence and low arousal categories), the Transformer GAN surpasses the simple Transformer due to the fact that it never situates more than 50% of the excerpts on the incorrect side of the middle line, which in this case is represented by the number 3. While the excerpts generated by the Compound Word Transformer also show this same characteristic, we see that there is not a strong reason for us to believe that one stands out in comparison to the other.

Overall, given the superior ratings of the Transformer GAN respective to the automatic metrics and its competitiveness with a state-of-the art model with respect to the human evaluations, and given the considerations above about model size and representation, the Transformer GAN seems to be a promising model for music generation conditioned by sentiment.

4.1 Conclusion and future work

We introduced a model capable of generating musical excerpts conditioned by labels that represent perceived emotion. Through the use of MLE pre-training and adversarial training, we guided our model towards understanding some aspects of the relationship between musical structure and affect. Our experiments show that both in terms of quality and the ability to communicate emotion, the samples generated by the proposed model achieve competitive results. Furthermore, our work points to several possible avenues for future research, such as the use of other symbolic representations of music, the development of generative models conditioned by emotion that work directly on audio, and further exploration of the use of affective conditioning signals, e.g., using them to guide automatic composition second-by-second or to automatically generate royalty free music notation based on specific sentiments.

	H	O	S	OQ
Baseline [37]	3.32 ± 1.29	2.93 ± 1.13	3.18 ± 1.30	3.49 ± 1.04
Transformer	3.75 ± 1.24	3.22 ± 1.19	3.76 ± 1.14	3.89 ± 1.14
Transformer-GAN	3.56 ± 1.34	3.06 ± 1.21	3.38 ± 1.09	3.44 ± 1.15

Table 2: Results of the Survey where participants were asked to rate the samples generated by several models. The columns are, respectively, Human-Likeness, Originality, Structure and Overall Quality.

5. REFERENCES

[1] Y. Kim, E. Schmidt, R. Migneco, B. Morton, P. Richardson, J. Scott, J. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, 01 2010.

[2] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

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

[4] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[5] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 1171–1179.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, 06 2014.

[7] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[8] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1erHoR5t7

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf

[10] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[11] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee

[12] M. J. Kusner and J. M. Hernández-Lobato, "Gans for sequences of discrete elements with the gumbel-softmax distribution," 2016.

[13] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. J. Smola, "Symbolic music generation with transformer-gans," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 408–417.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[16] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," 2021.

[17] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[18] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[19] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkgNKkHtvB

[20] K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller, "Rethinking attention with performers," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=Ua6zuk0WRH

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[22] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rJe4ShAcF7

[23] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 464–468. [Online]. Available: https://aclanthology.org/N18-2074

[24] C. Payne, ""musenet."," 2019.

[25] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *ISMIR*, 2017.

[26] H.-W. Dong and Y.-H. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *ISMIR*, 2018.

[27] N. Zhang, "Learning adversarial transformer for symbolic music generation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2020.

[28] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. [Online]. Available: https://aclanthology.org/P19-1285

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[30] R. Madhok, S. Goel, and S. Garg, "Sentimozart: Music generation based on emotions," in *ICAART*, 2018.

[31] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on lstm networks," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 2039–2043.

[32] X. Tan, M. Antony, and H. Kong, "Automated music generation for visual art through emotion." in *ICCC*, 2020, pp. 247–250.

[33] H. H. Tan and D. Herremans, "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*, 2020.

[34] L. Ferreira, L. Lelis, and J. Whitehead, "Computer-generated music for tabletop role-playing games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 59–65.

[35] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," *Proceedings of the Conference of the International Society for Music Information Retrieval*, 2019.

[36] D. Makris, K. R. Agres, and D. Herremans, "Generating lead sheets with affect: A novel conditional seq2seq framework," *arXiv preprint arXiv:2104.13056*, 2021.

[37] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proc. Int. Society for Music Information Retrieval Conf.*, 2021.

[38] W. Hsiao, J. Liu, Y. Yeh, and Y. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," *CoRR*, vol. abs/2101.02402, 2021. [Online]. Available: https://arxiv.org/abs/2101.02402

[39] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1180–1188. [Online]. Available: https://doi.org/10.1145/3394171.3413671

[40] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[41] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[42] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[43] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 2018*, 2018. [Online]. Available: https://arxiv.org/abs/1710.11153

[44] W. Nie, N. Narodytska, and A. Patel, "Relgan: Relational generative adversarial networks for text generation," in *International conference on learning representations*, 2018.

[45] L.-C. Yang and A. Lerch, “On the evaluation of generative models in music,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.

[46] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, “Muspy: A toolkit for symbolic music generation,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[47] P. N. Juslin, J. A. Sloboda *et al.*, “Music and emotion,” *Theory and research*, 2001.

IMPROVING CHORAL MUSIC SEPARATION THROUGH EXPRESSIVE SYNTHESIZED DATA FROM SAMPLED INSTRUMENTS

Ke Chen[1] **Hao-Wen Dong**[1] **Yi Luo**[2] **Julian McAuley**[1]
Taylor Berg-Kirkpatrick[1] **Miller Puckette**[1] **Shlomo Dubnov**[1]
[1] UC San Diego, USA [2] Tencent AI Lab, China
[1]{knutchen, hwdong, jmcauley, tberg, msp, sdubnov}@ucsd.edu [2]oulyluo@tencent.com

ABSTRACT

Choral music separation refers to the task of extracting tracks of voice parts (e.g., soprano, alto, tenor, and bass) from mixed audio. The lack of datasets has impeded research on this topic as previous work has only been able to train and evaluate models on a few minutes of choral music data due to copyright issues and dataset collection difficulties. In this paper, we investigate the use of synthesized training data for the source separation task on real choral music. We make three contributions: first, we provide an automated pipeline for synthesizing choral music data from sampled instrument plugins within controllable options for instrument expressiveness. This produces an 8.2-hour-long choral music dataset from the JSB Chorales Dataset and one can easily synthesize additional data. Second, we conduct an experiment to evaluate multiple separation models on available choral music separation datasets from previous work. To the best of our knowledge, this is the first experiment to comprehensively evaluate choral music separation. Third, experiments demonstrate that the synthesized choral data is of sufficient quality to improve the model's performance on real choral music datasets. This provides additional experimental statistics and data support for the choral music separation study.

1. INTRODUCTION

Choral music is a distinct artistic genre that includes several vocal parts (e.g. soprano, alto, tenor, and bass) arranged into intricate patterns from strict counterpoint to polyphonic echoes and flows of lyrics. One useful tool in the analysis and re-production of choral tracks is the ability to take mixed-down choral music and separate it back into audio tracks of isolated vocal parts: i.e. choral music separation, as a subtask of audio source separation.

Audio source separation is an audio signal processing task that involves separating one or more sound sources from a multi-source audio mixture. This task has a wide range of applications in a variety of domains, including speech separation, vocal-accompaniment separation and musical instrument separation. The latter two tasks are primary tasks in the field of music signal processing and have been adopted for practical use in the entertainment industry [1]. Many models, such as Open-Unmix [2], Demucs [3], and Spleeter [4], achieve great separation performance. Some models [5] further extend the source separation task to a zero-shot or query-based setting. However, choral music separation has received limited attention. Unlike general musical instrument separation, which seeks to separate non-homologous sources (e.g., piano, drums, and singing voice), choral music instrument separation seeks to separate homologous or close-homologous sources (e.g., soprano, alto, tenor, bass). Additionally, the scarcity of data on choral music separation impedes further progress. Choral music separation could be used in a wide variety of scenarios. Individuals could obtain solo tracks from choral recordings for practice, analysis, and re-production. Not only does it fill a void in a particular type of musical instrument separation, but it also provides convenience for music educators.

In this paper, we investigate the choral music separation task from the perspective of addressing the insufficiency of available datasets. We begin by introducing related works in the field of choral music separation. Second, we present the discovery of how to improve the performance of choral music separation using high-quality, synthesized music data. Then, we conduct comprehensive experiments with multiple models and datasets to evaluate the improvement of using synthesized data on choral music separation in real datasets. Finally, we discuss the extensibility of our pipeline to more choral-related separations, such as string quartet separation, as well as its future directions. The code and the dataset are publicly available [1] .

2. RELATED WORK

Research in choral music separation receives relatively less attention. Deep learning methods for audio source separation has already outperformed traditional methods (e.g. Non-negative Matrix Factorization [6]) for a long time. Separation models have been developed following two directions: frequency-domain models and time-domain separation models.

[1] https://github.com/RetroCirce/Choral_Music_Separation

Dataset	Minutes	Songs	Public
Choral Singing Dataset [11]	7	3	✓
Dagstuhl ChoirSet [12]	5	2	✓
Cantoria Dataset [13]	20	14	✓
ESMUC Choir Dataset [13]	31	26	✓
Bach and Barbershop Collection [14]	105	48	

Table 1: Existing datasets for choral music separation.

2.1 Frequency-Domain and Time-Domain Models

The traditional method of audio separation is to mask the frequency-domain representation and then inversely transform it to the time-domain signal, referred as frequency-domain separation models. Spec-U-Net [7], based on the U-Net architecture, contains convolutional neural network (CNN) blocks for downsampling the input short-time Fourier transform (STFT) spectrogram and upsampling the bottom feature back into a separation mask. The mask is applied into the input to obtain the separate spectrogram as output. Res-U-Net [8] replaces the original CNN blocks with residual CNN blocks to accelerate convergence speed. On the other hand, time-domain models perform the separation directly on the audio waveform. Wave-U-Net [9] incorporates an end-to-end U-Net structure on the input and output of waveforms. Conv-TasNet [10] applies a CNN encoder-decoder structure to process waveforms into latent features and generates the mask. The masked latent features are decoded back to waveforms as separation results. Bypassing the spectrogram processing, time-domain models can save parameters and perform efficiently in low-latency systems for speech separation. Some hybrid models, such as Demucs v3 [3], can leverage both time-domain and frequency-domain features to achieve the best performance for musical instrument separation, while the size of the model is a little bit large.

2.2 Choral Music Separation

For choral music separation, [15] proposed a score-informed separation model based on Wave-U-Net and performed experiments on 347 (synthesized) Bach Chorale pieces from MIDI files with *MuseScore_General* SoundFont. This model performs well on this SoundFont-Synthesis dataset but poorly on real choral music datasets. [16] proposed a conditional Spec-U-Net to optimize the separation performance by conditioning on the fundamental frequency contour. However, as mentioned in their paper, due to the lack of choral music datasets, the evaluation was conducted on only three songs with a total duration of seven minutes. [14] proposed a harmonic overlap score to increase the model's sensitivity to different choral voices, thereby improving performance. It made use of a relatively large dataset containing 105 minutes of Bach and Barbershop Collections, but this dataset is *not* publicly available due to copyright concerns, which prevents it from being open source. And indeed, 100-minute is still not enough to help achieve an audio separation model with a high generalization ability, we expect to obtain a size more than that.

		Pitch range			
Name	Type	Soprano	Alto	Tenor	Bass
Standard MIDI		A0–C8	A0–C8	A0–C8	A0–C8
Noire [17]	Piano	A0–C8	A0–C8	A0–C8	A0–C8
Grandeur [18]	Piano	A0–C8	A0–C8	A0–C8	A0–C8
Voices Of Rapture [19]	Vocal	B3–D6	E3–G5	B2–C#5	A1–D4
Dominus Choir [20]	Vocal	G3–A5	G3–A5	E2–G4	E2–G4

Table 2: Sample instrument libraries we use for synthesizing choral music separation datasets.

In this paper, we first conduct four fundamental models: Spec-U-Net, Res-U-Net, Wave-U-Net and Conv-TasNet. Our objective is to demonstrate the efficacy of synthesized expressive data in improving separation performance on **real choral music datasets**. As a result, fundamental models enable us to consider the performance gains more directly associated with data changes and augmentations. Also, score-informed and conditional separation models introduce external information, such as musical notes of original songs or multi-pitch estimation results, to guide the separation's goal, while it also limits its applications. In practice, we frequently find ourselves in situations where the only available input is the audio. We continue to demand **unconditioned choral music separation**. As a result, we proceed directly to unconditioned choral music separation in this paper, without relying on any score conditions.

3. METHODOLOGY

3.1 Scarcity of Datasets

Existing datasets for choral music are listed in Table 1, collected from previous works and other public sources. We observe that most of these datasets have short total lengths; three of them are less than 20 minutes. The Choral Singing Dataset [11] and ESMUC Choir Dataset [13] have been used for choral music separation by [16], while the Dagstuhl ChoirSet [12] and Cantoria [13] Datasets were never used for separation tasks but instead for singing performance analysis. The Bach and Barbershop Collection [14] is relatively longer, but is not publicly available. As a result, when a model is trained and tested on such a small amount of data, its generalization and separation capabilities are severely limited. [15] directly synthesized 347 choral pieces of Bach's from MIDI files with *MuseScore_General* SoundFont and trained the model. However, this SoundFont-Synthesis dataset is dissimilar to true choral vocals. Moreover, it lacks lyrics and syllables. As a result, the trained model performs poorly on real datasets [15].

In the next section, we first introduce the pipeline of synthesizing audio datasets for choral music separation from sampled instrument libraries. Then, we train various models on our datasets and compare them to determine the best model. Finally, we transfer the best model weights to the real-world datasets shown in Table 1, fine-tune the model and determine whether it truly improves