groups. Since the piano is a fixed-pitch keyboard instrument, the fundamental frequency of each key does not vary as time changes. This enables the model process each frequency group independently. Based on this assumption, we feed each frequency group to the same LSTM layer individually to model the temporal relationship. We term this the Frequency Grouped LSTM (FG-LSTM) illustrated in Figure 1. Feeding a single frequency group to the LSTM makes inputs and units size of FG-LSTM smaller. This method significantly reduces the amount of parameters in our model.

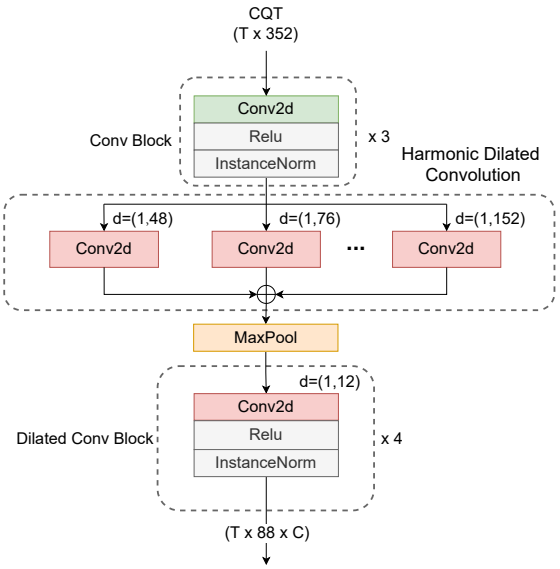

Figure 2. Acoustic model of HPPNet. It is composed of three identical regular convolution layers, a harmonic dilated convolution layer, a max pooling layer, and four identical dilated Conv blocks. Different dilated convolution layers have different dilation rates denote by d. The output is a feature map with size $T \times 88 \times C$, where T is the frame numbers, 88 denotes the numbers of piano keys, C is the channel size.

2.4 Model Details

Raw audios are clipped to 20-second pieces and resampled to 16 kHz. Then, the CQT is computed by nnAudio [27] with hop length 320 (20 millisecond), an FFT window of 2048, bins per octave of 48, fmin of 27.5 Hz, frequency bins number of 352, and log amplitude. The model takes the CQT feature with size $T \times 352$ as input, where T is the frame number of each audio clip. The time resolution and frequency resolution of inputs are limited by the computational resource, higher input resolution may have better performance.

The model is composed of a convolutional acoustic model and multiple FG-LSTM heads. The former structure captures the harmonic representations of a tone, and the latter establishes connections over temporal series. In the acoustic model shown in Figure 2, the first three convolution layers extract local information with kernel size 7×7, a ReLu activation, and instance normalization.

Then followed by eight dilated convolution layers parallel with different dilations of 48, 76, 96, 111, 124, 135, 144, and 152 in the frequency dimension. These dilation rates are calculated by Eq. (1) with $Q = 48$ and rounded to integers. All the outputs of these layers integrate the harmonic information for each frequency unit.

The max-pooling layer downsamples the frequency bins from 352 to 88 with pooling size of 4, four bins per semitone to one bin per semitone. Then four identical dilated convolution blocks make the network deeper to achieve higher learning capacity. Each block contains one convolution layer with kernel size of 5×3, and dilation of 1×12.

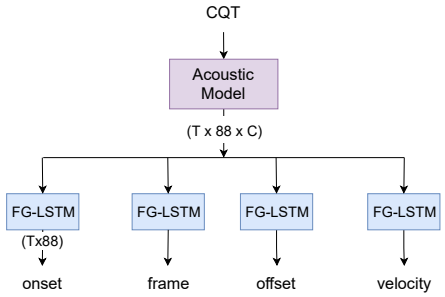

Figure 3. HPPNet-base. Onset, frame, offset, and velocity tasks share the same acoustic model.

Finally, four FG-LSTM heads are used to predict frames, onsets, offsets, and velocities separately as shown in Figure 3. These outputs are then decoding to note events with the same way as the Onsets & Frames [8] model. Note that the offset head is just used for training and not directly used during decoding. Each head uses a linear layer with a sigmoid activation function to output the prediction. All the outputs are matrices of size $T \times 88$, where T denotes frame number and 88 represents the 88 piano keys. The LSTM is bidirectional, and both forward and backward directions have 64 units. The hyperparameters of the network are summarized in Table 2. We denote this model the HPPNet-base.

Layer	repeat	kernel	filters	dilation
Conv	3	7×7	16	-
HD-Conv	1	1×3	128	48, 76, ..., 152
Dilated Conv	4	5×3	128	1×12

Table 2. Hyperparameters of convolution layers.

All the FG-LSTM heads in HPPNet-base share the same acoustic model, making it simple and efficient. But in our experiments, we try multiple acoustic models and find that if a stand-alone acoustic model is used to predict the onset, the model will perform better in onset detection. Since the onset F-measure is the metric that correlates best with human judgment [28], we use an individual acoustic model for onset. Frame, offset, and velocity share another acoustic model. The HPPNet-sp is shown in Figure 4. The output representations of the onset acoustic model are concatenated with outputs of the other acoustic model to pre-

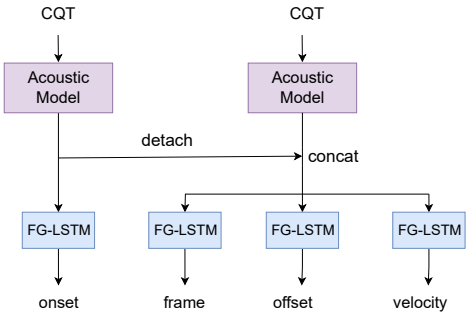

Figure 4. HPPNet-sp. Onset and other tasks use separate acoustic models.

dict frame, offset, and velocity (detach to avoid backward propagation).

3. EXPERIMENTS

In this section we first introduce the datasets we used. And then, we describe the details of the experiment and the training loss for our model.

3.1 Datasets

MAESTRO [8] (MIDI and Audio Edited for Synchronous Tracks and Organization): It includes about 200 hours of paired audio and MIDI recordings from ten years of International Piano-e-Competition. Audio and MIDI files are aligned with 3 ms accuracy and sliced to individual musical pieces annotated with composer, title, and year of performance. The MIDI have been collected by Yamaha Disklaviers which have a high-precision MIDI capture system.

MAPS [29] (MIDI Aligned Piano Sounds): It includes about 31 GB of CD-quality recordings and corresponding annotations of isolated notes, chords, and complete piano pieces. The records contain both synthesized audio and real audio by Virtual Piano software and a Yamaha Disklavier respectively.

3.2 Experimental setup

We use PyTorch [1] to implement our model. The training is performed by Adam [30] optimizer with mini-batch size 4, and a learning rate of 0.0006 for 200k to 500k steps with early stopping. The training time on an NVIDIA GeForce 3060 GPU with 12 GB VRAM is about 48 hours. The note and frame scores are calculated by the mir_eval library [31]. The evaluation uses a 50 ms tolerance for note onset and an offset ratio of 0.2 as the default configuration in mir_eval. The onset and frame thresholds are both set to 0.4 on the sigmoid output in our model. All the hyperparameters are selected by the performance on the validation split of MAESTRO.

The losses we use for training are similar to the Onsets & Frames model. The losses of onset, frame, and offset are binary cross-entropy losses. The weighted frame loss is not used in our training. The loss of velocity is the mean squared error loss. All losses are summed together as the final loss. All the losses are as follows:

$$l_{bce}\left(y\ ,\ \hat{y}\right) = -wy\log(\hat{y}) - (1-y)\log(1-\hat{y}), \quad (2)$$

$$L_{onset} = \sum_{p=1}^{88}\sum_{t=1}^{T} l_{bce}\left(n_{p,t}\ ,\ \hat{n}_{p,t}\right), \quad (3)$$

$$L_{frame} = \sum_{p=1}^{88}\sum_{t=1}^{T} l_{bce}\left(f_{p,t}\ ,\ \hat{f}_{p,t}\right), \quad (4)$$

$$L_{offset} = \sum_{p=1}^{88}\sum_{t=1}^{T} l_{bce}\left(o_{p,t}\ ,\ \hat{o}_{p,t}\right), \quad (5)$$

$$L_{velocity} = \sum_{p=1}^{88}\sum_{t=1}^{T} n_{p,t}\left(v_{p,t} - \hat{v}_{p,t}\right)^2, \quad (6)$$

$$L = L_{onset} + L_{frame} + L_{offset} + L_{velocity}. \quad (7)$$

where l_{bce} denotes the binary cross entropy loss, p is the piano note index range from 1 to 88, T is the frame number in a sample, w denotes the positive weight ($w = 2$ for onset, $w = 1$ for frame and offset), y denotes ground true and $\hat{y}$ denotes predicted values range from 0 to 1, $n_{p,t}$, $f_{p,t}$, $o_{p,t}$, and $v_{p,t}$ are the ground true of onset, frame, offset, and velocity seperately.

3.3 Baselines

Our model is compared with five typical models, including High-Resolution piano transcription [12], Onsets & Frames [9], Adversarial onsets & frames [11], Semi-CRFs [32] and the sequence-to-sequence Transformer [10]. The results are shown in Table 3. High-Resolution and Onsets & Frames are convolution-based models composed of convolution layers and LSMT layers. Especially, the High-Resolution is the best model in note F1 score which is 96.72%. Adversarial onsets & frames is an adversarial training scheme that operates on the Mel-spectrogram. Semi-CRFs is a Semi-Markov Conditional Random Fields (semi-CRF) based model, which treats note intervals as holistic events. Sequence-to-sequence Transformer uses a generic Transformer architecture with standard decoding methods.

4. RESULTS

We compare our model with some existing state-of-the-art methods using the MAESTRO dataset and the MAPS dataset. Then, ablation studies are done to demonstrate the affects of FG-LSTM, and HD-Conv. We also examine the model performance on small datasets.

[1] www.pytorch.org

Model	Params	FRAME			NOTE			NOTE W/OFFSET			NOTE W/OFFSET & VEL.		
		P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
		MAESTRO v1											
Onsets & Frames [8]	26M	92.11	88.41	90.15	98.27	92.61	95.32	82.95	78.24	80.50	79.89	75.37	77.54
Adversarial onsets & frames [11]	26M	93.1	89.8	91.4	98.1	93.2	95.6	83.5	79.3	81.3	82.3	78.2	80.2
		MAESTRO v2											
High-Resolution [12]	20M	88.71	90.73	89.62	98.17	95.35	96.72	83.68	81.32	82.47	82.10	79.80	80.92
Semi-CRFs [32]	9M	93.85	88.72	91.11	98.66	94.50	96.51	90.68	86.89	**88.72**	89.68	85.96	**87.75**
HPPNet-sp	1.2M	92.36	93.46	**92.86**	98.31	96.18	**97.21**	85.36	83.54	84.41	83.85	82.08	82.93
		MAESTRO v3											
Onsets & Frames [reproduced]	26M	94.43	85.50	89.68	98.67	92.08	95.22	82.25	76.88	79.44	80.80	75.55	78.05
Transformer [10]	54M	-	-	-	-	-	96.13	-	-	83.94	-	-	82.75
Semi-CRFs [32]	9M	93.79	88.36	90.75	98.69	93.96	96.11	90.79	86.46	**88.42**	89.78	85.51	**87.44**
HPPNet-tiny	151K	92.26	91.98	92.06	96.75	94.94	95.82	83.77	82.26	83.00	82.77	81.29	82.01
HPPNet-base	820K	92.35	92.75	92.51	97.60	94.90	96.25	84.03	83.48	83.57	82.94	81.23	82.12
HPPNet-sp	1.2M	92.79	93.59	**93.15**	98.45	95.95	**97.18**	84.88	82.76	83.80	83.29	81.24	82.24

Table 3. Transcription result evaluated on the MAESTRO dataset. Train split of MAESTRO v3 is used for training and test splits of different versions are used for evaluation.

Model	Params	FRAME			NOTE			NOTE W/OFFSET			NOTE W/OFFSET & VEL.		
		P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
Onsets & Frames	26M	-	-	82.02	-	-	83.04	-	-	61.84	-	-	48.07
Onsets & Frames*	26M	92.86	78.46	84.91	87.46	85.58	86.44	68.22	66.75	67.43	52.41	51.22	51.77
HPPNet-sp	1.2M	88.42	86.81	87.56	91.61	82.38	86.63	65.01	63.84	64.39	60.35	59.26	59.77

Table 4. The transcription evaluation result on MAPS test split, which training was done on the MAESTRO v3 train split. Onsets & Frames* means Onsets & Frames with audio augmentation. The training of HPPNet is without audio augmentation.

4.1 Comparison with baselines

Along with the HPPNet-base and HPPNet-sp, we also evaluate the HPPNet-tiny, which has a similar structure as HPPNet-base but reduces the maximum convolution channel size and units in LSTM from 128 to 48.

Evaluation on the MAESTRO dataset is shown in Table 3. The HPPNet-sp achieves state-of-the-art in both frame F1 score and note F1 score, which improved 3.24 percentage points and 0.49 percentage points, reaching 92.86% and 97.21% in MAESTRO v2 respectively. In the last two columns, under the condition of note with offset, it still outperforms other models except the event-based Semi-CRFs method. Even slight structure of HPPNet-base still achieves 92.51% and 96.25% on frame F1 score and note F1 score, respectively.

Moreover, another significant advantage is that our model has much fewer parameters, around 820 thousand HPPNet-base and 1.2 million HPPNet-sp. The lightweight structure HPPNet-tiny with 151 thousand parameters still outperforms the baseline Onsets & Frames, that the frame F1 score and the note F1 score reaching 92.06% and 95.82%. This proves the effectiveness of HPPNet in both improving performance and reducing model size.

Empirical results in Table 3 show that among all compared frame-level models, our model performs best in all scores. It is interesting to note that HPPNet achieves improved recall while not or only slightly loosing precision. But the event-based Semi-CRFs has better prediction on offsets, for it predicts note intervals directly rather than individual frames. Maybe future works that combining the frame-level and event-level can take advantage of the both sides.

To further explore the generalization of the HPPNet in different datasets, we train the model on the MAESTRO dataset and evaluate it on the MAPS dataset. The result is shown in Table 4. we can find that the HPPNet-sp's note F1 still reaches 87.56%. This result is better than Onsets & Frames of 83.04%, even outperforming Onsets & Frames with audio augmentation of 86.44%. The result confirms that the HPPNet has better generalization ability on piano transcription task.

4.2 Ablation study

We further analyze the role of the HD-Conv layer and FG-LSTM in the HPPNet. In addition to the HPPNet, we evaluate three types of models, the F1 in frame-wise

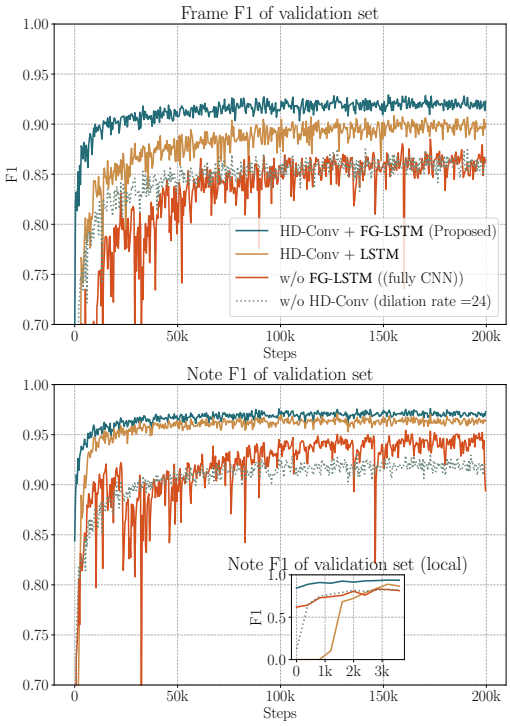

Figure 5. F1 in frame-wise and note-wise of MAESTRO v3 validation set.

and note-wise during training per epoch shown in Figure 5:

- To verify the effect of FG-LSTM: change FG-LSTM to normal LSTM with frequency flattening.
- To verify the effect of LSTM: remove time-series layers, which change them to linear layers.
- To verify the effect of harmonic dilated convolution: replace the harmonic dilated convolution with fixed dilated convolution (dilation rate of 24).

These results suggest that the the HPPNet outperforms the other three ablated models. The F1 of normal LSTM decreases about three percentage points and one percentage point on frame-wise and note-wise, respectively. Usual LSTM starts later than the HPPNet in note-wise F1 at the beginning of training, and it starts to increase after about 1k steps. The HD-Conv + vanilla LSTM is better than changing time-series layers to linear layers. Model without RNN converges slowly and fluctuates drastically. A large receptive field helps model get global information. Using the fixed dilation rate has a similar receptive field to HD-Conv. But the result shows it does not capture useful information for the detection of frames and onsets, leading to poor performance.

4.3 Performance on small datasets

The HPPNet trained on big dataset such as compete MAESTRO shows promising results, and it also shows better generalization when inference on MAPS. To evaluate the performance of the HPPNet on smaller dataset, we also train the model with 30% and 10% of MAESTRO training split. All the samples are randomly selected and without data augmentations. The results are displayed in Table 5. When the training set size decreases to 10%, the HPPNet-sp drops less than Onsets & Frames on frame F1 and note F1 scores with 88.31% and 93.52% respectively, while the Onsets & Frames approach only yields 80.36% and 90.62%. This demonstrates that the HPPNet relies less on data thanks to the small amount of parameters and priors contained in HD-Conv and FG-LSTM.

Model	Data	FRAME			NOTE		
		P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
Onsets& Frames	100%	94.43	85.50	89.68	98.67	92.08	95.22
	30%	91.14	80.00	85.08	98.13	89.57	93.60
	10%	90.66	72.70	80.36	96.64	85.46	90.62
HPPNet-sp	100%	92.79	93.59	**93.15**	98.45	95.95	**97.18**
	30%	90.72	92.11	91.35	96.46	94.46	95.43
	10%	92.10	84.96	88.31	96.59	90.94	93.52

Table 5. The transcription evaluation result on MAESTRO v3 test split, which training was done on the 100%, 30%, and 10% of MAESTRO v3 training split respectively. Considering the annotation accuracy, we use subsets of MAESTRO rather than the MAPS for evaluation.

5. CONCLUSION

In this paper, we design a model that carries piano inductive biases to capture piano priors. The HPPNet is proposed to model the harmonic structure and the pitch-invariance over time in piano transcription. Experimental results show that the model achieves state-of-the-art performance on the MAESTRO dataset, with frame F1 of 93.15% and note F1 of 97.18%. The success stems from two aspects: (i) the harmonic dilated convolution exploited to capture harmonics structure; and (ii) the frequency grouped LSTM designed based on the pitch-invariance of piano key over time. Furthermore, the model parameters are much fewer than the previous state-of-the-art models.

The results of the model on piano transcription are encouraging. And it may also be suitable for other instruments with similar pitch characteristics. But some challenges remain, such as improving performance in offset detection and modifying the model to other polyphonic transcriptions involving vocals and instruments with a less stable pitch. The harmonic dilated convolution implemented by multiple parallel dilated convolutions is computational consuming and needs to be optimized in the future.

6. ACKNOLEDGEMENT

This work was supported by National Key R&D Program of China(2019YFC1711800), NSFC(62171138). Wei Li and Yi Yu are corresponding authors of this paper.

7. REFERENCES

[1] A. Khlif and V. Sethu, "An iterative multi range non-negative matrix factorization algorithm for polyphonic music transcription," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, 2015, pp. 330–335.

[2] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

[3] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Speech Audio Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[4] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A classification-based polyphonic piano transcription approach using learned feature representations," in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR*, 2011, pp. 175–180.

[5] T. Cheng, M. Mauch, E. Benetos, and S. Dixon, "An attack/decay model for piano transcription," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR*, 2016, pp. 584–590.

[6] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2012, pp. 121–124.

[7] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE ACM Transactions on Audio Speech and Language Processing. TASLP*, vol. 24, no. 5, pp. 927–939, 2016.

[8] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations, ICLR*, 2019, pp. 1–6.

[9] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. 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, pp. 50–57.

[10] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 246–253.

[11] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 670–677.

[12] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE ACM Transactions on Audio Speech and Language Processing. TASLP*, vol. 29, pp. 3707–3717, 2021.

[13] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America, JASA*, vol. 89, no. 1, pp. 425–434, 1991.

[14] W. Wei, P. Li, Y. Yu, and W. Li, "Harmof0: Logarithmic scale dilated convolution for pitch estimation," in *2022 IEEE International Conference on Multimedia and Expo, ICME*, 2022, pp. 1–6.

[15] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 estimation in polyphonic music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, 2017, pp. 63–70.

[16] Z. Zhang, Y. Wang, C. Gan, J. Wu, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, "Deep audio priors emerge from harmonic convolutional networks," in *8th International Conference on Learning Representations, ICLR*, 2020, pp. 1–8.

[17] X. Wang, L. Liu, and Q. Shi, "Enhancing piano transcription by dilated convolution," in *19th IEEE International Conference on Machine Learning and Applications, ICMLA*, 2020, pp. 1446–1453.

[18] X. Wang, L. Liu, and Q. Shi, "Harmonic structure-based neural network model for music pitch detection," in *19th IEEE International Conference on Machine Learning and Applications, ICMLA*, 2020, pp. 87–92.

[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of International Conference on Learning Representations, ICLR*, 2016, pp. 1–4.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, pp. 2673–2681, 1997.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*, 2017, pp. 21–25.

[23] X. Li and R. Horaud, "Multichannel speech enhancement based on time-frequency masking using subband long short-term memory," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*, 2019, pp. 298–302.

[24] B. Wang and Y.-H. Yang, "Performancenet: Score-to-audio music generation with multi-band convolutional residual network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1174–1181.

[25] Y. Luo, C. Han, and N. Mesgarani, "Ultra-lightweight speech separation via group communication," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2021, pp. 16–20.

[26] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band melgan: Faster waveform generation for high-quality text-to-speech," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 492–498.

[27] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "Nnaudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1d convolutional neural networks," *IEEE Access*, vol. 8, pp. 161 981–162 003, 2020.

[28] A. Ycart, L. Liu, E. Benetos, and M. T. Pearce, "Investigating the perceptual validity of evaluation metrics for automatic piano music transcription," *Transactions of the International Society for Music Information Retrieval Conference TISMIR*, pp. 68–81, 2020.

[29] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE ACM Transactions on Audio Speech and Language Processing. TASLP*, vol. 18, no. 6, pp. 1643–1654, 2009.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations, ICLR*, 2015, pp. 1–9.

[31] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "Mir_eval: A transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*, 2014, pp. 1–6.

[32] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural semi-crfs," in *Advances in Neural Information Processing Systems, NeurIPS*, vol. 34, 2021, pp. 20 583–20 595.

GENERATING MUSIC WITH SENTIMENT USING TRANSFORMER-GANS

Pedro L. T. Neves
State University of Campinas
p185770@dac.unicamp.br

José E. F. Novo Junior
State University of Campinas
fornari@unicamp.br

João B. Florindo
State University of Campinas
florindo@unicamp.br

ABSTRACT

The field of Automatic Music Generation has seen significant progress thanks to the advent of Deep Learning. However, most of these results have been produced by unconditional models, which lack the ability to interact with their users, not allowing them to guide the generative process in meaningful and practical ways. Moreover, synthesizing music that remains coherent across longer timescales while still capturing the local aspects that make it sound "realistic" or "human-like" is still challenging. This is due to the large computational requirements needed to work with long sequences of data, and also to limitations imposed by the training schemes that are often employed. In this paper, we propose a generative model of symbolic music conditioned by data retrieved from human sentiment. The model is a Transformer-GAN trained with labels that correspond to different configurations of the valence and arousal dimensions that quantitatively represent human affective states. We try to tackle both of the problems above by employing an efficient linear version of Attention and using a Discriminator both as a tool to improve the overall quality of the generated music and its ability to follow the conditioning signals.

1. INTRODUCTION

One of the driving factors behind the human experience of music is the emotional content that it conveys. Several works in the area of Musical Information Retrieval (MIR) have focused on Music emotion recognition, that is, automatic recognition of the perceived emotion of music based solely on the musical information itself [1]. In this context, emotion is often represented according to the valence and arousal dimensions originated from to the Russell circumplex model [2]. While valence expresses how pleasurable or displeasurable an emotion is, arousal represents the level of alertness associated with that emotion, going from relaxed to excited. However, within the realm of *Deep Learning* comparatively little research has focused on reverting this process, that is, instead of predicting the affective content of a musical passage, being able to generate musical pieces that project a desired emotional state.

Deep Neural Networks are now capable of generating songs that display coherent chord progressions, melodies and even lyrics [3], despite the fact that these characteristics often do not persist across longer time scales. While there are several reasons behind this fact, two of them stand out. Firstly, there are the inherent computational and memory costs involved in modeling longer sequences of data. Secondly, the most common technique used to train these models, teacher forcing or Maximum Likelihood Estimation (MLE) [4], makes them work with data distributions that are different during training and inference time (real vs synthetic). This is known as exposure bias [5]. One of the ways to alleviate this problem is by delegating the task of judging which samples are good and which are not to a different neural network that is trained in conjunction with the generative model. This is the motivation behind the use of Generative Adversarial Networks (GANs) [6] within the realm of sequence generation [7].

In this work, we present a generative model of music capable of synthesizing songs conditioned by values of valence and arousal that correspond to perceived sentiment [2]. The ability of automatically generating music that follows a specific pattern of emotions can be interesting in various contexts, e.g. producing soundtracks to accompany story-driven forms of media such as Video-Games and Movies, which often use music as a means of guiding the audience towards a specific emotional state that suits the narrative. The goal here is to provide users who may not have the musical background necessary for composition a way of translating their perception into songs that can suit their artistic aspirations.

In an effort to mitigate the shortcomings of teacher forcing-style training, we complement it with an additional adversarial signal provided by a Discriminator network. This addition has a positive effect on the generated samples, and we demonstrate, through evaluations of our model both via automatic metrics and human feedback, that the proposed Transformer GAN obtains a performance that competes with a current state-of-the-art model, even while having a smaller set of parameters and using a simpler representation of music. To summarise, our contributions are as following:

- We present a neural network that, up to our knowledge, is the first generative model based on GANs to produce symbolic music conditioned by sentiment.

- We show, both through automatic and human evaluation, that our model obtains a competitive performance with that of a current state-of-the-art model on the task of music generation conditioned by sentiment.
- We show that promising results come from using Generative Adversarial Networks within the context of music generation conditioned by sentiment, as our model obtains a good performance despite having less parameters, using a simpler symbolic representation, and, being, up to our knowledge, the first on this task to be trained via an adversarial scheme.

2. RELATED WORKS

2.1 Generative Adversarial Networks

Generative Adversarial Networks, or simply GANs, consist of a theoretical framework in which two Neural Networks, the Generator and the Discriminator, through competition, optimize a model that implicitly approximates a data distribution by generating samples that try to mimic the features that it observes on a given set of samples originating from that distribution. Each of the networks is trained to optimize an objective function. The objective of the Discriminator D is to separate the real samples from those that are created by the Generator G, whose job is to produce samples that are so similar to those of the real distribution that the Discriminator is unable to determine which of them are real and which are fake. This whole process is equivalent to a min-max game that can be formalized by Equation 1.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

In the equation above, $p_{\text{data}}(\boldsymbol{x})$ is the real data distribution, and $p_z(\boldsymbol{z})$ is a prior on input noise variables that come from a normal distribution, and which are then mapped to data space by G, so that the synthetic distribution, p_g, can be learned.

Due to the inherent instability of the adversarial process, several works have focused on improving the convergence and the quality of the samples generated by GANs via new objective functions, regularization and normalization techniques, and model architectures. Of particular interest to this work is RSGAN [8], which substitutes the standard GAN loss for the non-saturating Relativistic Standard Loss. Here, as the authors put it, the Discriminator estimates the probability that the given real data is more realistic than a randomly sampled fake data. Equations 2 and 3 correspond to the objectives for the Discriminator and the Generator, respectively.

$$\mathcal{L}_{RSGAN,D} = - \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})}[\log(\text{sigmoid}(D(x_r) - D(x_f)))] \quad (2)$$

$$\mathcal{L}_{RSGAN,G} = - \mathbb{E}_{(x_r,x_f)\sim(\mathbb{P},\mathbb{Q})}[\log(\text{sigmoid}(D(x_f) - D(x_r)))], \quad (3)$$

in which $\mathbb{P}$ and $\mathbb{Q}$ are, in this order, the real and fake disributions.

In order to provide more stability to the training process, WGAN-GP [9] introduces a Gradient Penalty in the form of an additional loss that enforces a Lipschitz constraint on the Discriminator. This loss is expressed in Equation 4:

$$\mathcal{L}_{GP} = \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D_\phi(\hat{x})\|_2 - 1)^2] \quad (4)$$

where $\hat{x}$ is sampled along straight lines between pairs of points taken from the real and fake data distributions, and ϕ are the Discriminator parameters.

One possible strategy that can be used to combat the effect known as exposure bias [5], characterized by the disparity between the data available to the network during training and inference time when the model is trained via standard Maximum Likelihood Estimation (MLE), is the insertion of a Discriminator into the training process as a tool to guide the Generator. Nevertheless, generating discrete sequences with GANs is notoriously hard. This is mostly due to the fact that the outputs of some sequence models are discretized in a way that prohibits the gradient of the Discriminator loss to propagate through the Generator. Several strategies have been proposed to circumvent this issue [10–12]. Here, we employ the Gumbel-Softmax technique presented in [11] and applied to adversarial training in [12]. Mathematically, if we have a categorical distribution with class probabilities $\pi_i, i \in i, ..., d$, we can draw samples y from this distribution using:

$$y = \text{one\_hot}\left(\arg\max_i [g_i + \log \pi_i]\right) \quad (5)$$

where each $g_i, i \in i, ..., d$ is taken from a Gumbel Distribution with location 0 and scale 1. From this expression, one can obtain a continuously differentiable approximation of the categorical distribution parameterized in terms of the softmax function:

$$y = \text{softmax}((1/\tau)(\pi + g)), \quad (6)$$

where τ is a temperature parameter that regulates how close to the categorical (lower τ) *versus* to the uniform distribution (higher τ) is y. This parameter is annealed from large values to close to zero during training. Here, we use the same schedule as in [13], that is, $1/\tau = (1/\tau_{\min})^{n/N}$, where n is the index of the current global optimization, N is the total number of steps and $\tau_{\min}$ is a hyperparameter which we chose to be 10^{-2}.

2.2 Transformers

Shortly after its presentation in 2017, the Transformer [14] became the most popular architecture in the field of Natural Language Processing (NLP), and it has also been successfully applied to other areas, such as image recognition [15] and audio [16]. The main reason behind its success is the