AUTOMATIC MUSIC PRODUCTION USING GENERATIVE ADVERSARIAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

When talking about computer-based music generation, two are the main threads of research: the construction of *autonomous music-making systems*, and the design of *computer-based environments to assist musicians*. Despite consistent demand from music producers and artists, however, little effort has been done in the field of automatic music arrangement in the audio domain. In this work, we propose a novel framework for *automatic music arrangement from raw audio in the frequency domain*. Using several songs converted into Mel-spectrograms – a two-dimensional time-frequency representation of audio signals – we were able to automatically generate original arrangements for both bass and voice lines. Treating music pieces as images (Mel-spectrograms) allowed us to reformulate our problem as an *unpaired image-to-image translation* problem, and to tackle it with CycleGAN, a well-established framework. Moreover, the choice to deploy raw audio and Mel spectrograms enabled us to more effectively model long-range dependencies, to better represent how humans perceive music, and to potentially draw sounds for new arrangements from the vast collection of music recordings accumulated in the last century. Our approach was tested on two different downstream tasks: given a bass line creating credible and on-time drums, and given an acapella song arranging it to a full song. In absence of an objective way of evaluating the output of music generative systems, we also defined a possible metric for the proposed task, partially based on human (and expert) judgement. To the best of our knowledge, we are the first to address the music arrangement task in the audio domain, to treat music pieces as images, and to propose a quantitative approach to evaluate the model results.

1 INTRODUCTION

The development of home music production has brought significant innovations into the process of pop music composition. Software like Pro Tools, Cubase, and Logic – as well as MIDI-based technologies and digital instruments – allow artists and producers to easily manipulate recordings and create high quality songs directly from home. After recording a melody, maybe with the aid of a guitar or a piano, song writers can now start building up the arrangement one piece at a time, sometimes not even needing professional musicians or proper music training. As a result, singers and song writers – as well as producers – have started asking for tools that could facilitate, or to some extent even automate, the creation of full songs around their lyrics and melodies.

To meet this new demand, the goal of designing computer-based environments to assist human musicians has become central in the field of automatic music generation (Briot et al., 2020). IRCAM OpenMusic (Assayag et al., 1999), Sony CSL-Paris FlowComposer (Papadopoulos et al., 2016), and Logic Pro X Easy Drummer are just some examples. In addition, more solutions based on deep learning techniques, such as RL-Duet (Jiang et al., 2020) – a deep reinforcement learning algorithm for online accompaniment generation – or PopMAG, a transformer-based architecture which relies on a Multi-track MIDI representation of music (Ren et al., 2020), continue to be studied.

Most of these strategies, however, suffer from the same critical issue, which makes them less appealing in view of music production for commercial purposes: they rely on a symbolic/MIDI representation of music. The approach proposed in this paper, instead, is a first attempt at automatically generating an euphonic arrangement (two or more sound patterns that produce a pleasing and har-

monious piece of music) in the audio domain, given a musical sample encoded in a two-dimensional time-frequency representation (known as Mel-spectrogram). Although arrangement generation has been studied in the context of symbolic audio, indeed, switching to Mel-spectrograms allows us to preserve the sound heritage of other musical pieces (allowing operations such as sampling) and is more suitable for real-life cases, where voice, for instance, cannot be encoded in MIDI.

We focused our attention on two different tasks of increasing difficulty: (i) given a bass line to create credible and on-time drums, and (ii) given the voice line, to output a new and euphonic musical arrangement. Incidentally, we found out that – for training samples – our model was able to reconstruct the original arrangement pretty well, even though no pairing among the Mel-spectrograms of the two domains was performed.

By means of the Mel-spectrogram representation of music, we can consider the problem of automatically generating an arrangement or accompaniment for a specific musical sample equivalent to an image-to-image translation task. For instance, if we have the Mel-spectrogram of an acapella song, we may want to produce the Mel-spectrogram of the same song including a suitable arrangement. To solve this task, we tested an unpaired image-to-image translation strategy known as CycleGAN (Zhu et al., 2017), which consists of translating an image from a source domain X to a target domain Y in the absence of paired examples, by training both the mapping from X to Y and from Y to X simultaneously, with the goal of minimizing a cycle consistency loss.

The aforementioned system was trained on 5s pop music samples (equivalent to 256×256 Mel-spectrograms) coming both from the Free Music Archive (FMA) dataset (Defferrard et al., 2016), and from the Demucs dataset (Défossez et al., 2019). The short sample duration does not affect the proposed methodology, at least with respect to the arrangement task we focus on, and inference can be performed also on full songs. Part of the dataset was pre-processed first, since the FMA songs lacked source separated channels (i.e. differentiated vocals, bass, drums, etc.). The required channels were extracted using Demucs (Défossez et al., 2019).

The main innovations presented in this contribution are as follows:

- treating music pieces as images, we developed a framework to automatically generate music arrangement from raw audio in the frequency domain, different from any other previous approach;
- our approach is able to generate arrangements with low computational resources and limited inference time, if compared to other popular solutions for automatic music generation (Dhariwal et al., 2020);
- we developed a metric – partially based on or correlated to human (and expert) judgement – to automatically evaluate the obtained results and the creativity of the proposed system, given the challenges of a quantitative assessment of music.

To the best of our knowledge, this is the first work to face the automatic arrangement production task in the audio domain, thus in view of potential music production for commercial purpose, by leveraging a two-dimensional time-frequency representation.

2 Related Works

Automatic music generation. The interest surrounding automatic music generation has greatly increased in the last few years, as proven by the number of deep learning approaches proposed. Most of them aim at generating new music from scratch or at genre and instrument translation, and are based on Recurrent Neural Networks (Mehri et al., 2016; Docevski et al., 2018; Manzelli et al., 2018; Vasquez & Lewis, 2019; Jiang et al., 2019), Dilated Convolutional Neural Networks (Oord et al., 2016), Transformers (Dhariwal et al., 2020) or Generative Adversarial Networks (Dong et al., 2017; Yang et al., 2017; Kumar et al., 2019; Oza et al., 2020).

Moreover, we can observe a great variety in terms of music domains (waveforms, spectrograms/frequency) and ways of representing music – such as raw audio, MIDI (Dong et al., 2017; Yang et al., 2017; Zhu et al., 2018; Manzelli et al., 2018), piano rolls (Docevski et al., 2018; Jiang et al., 2019), music sheets etc. In particular, raw audio generation is common to several sub-fields, such as speech synthesis and music generation (Zhao et al., 2020); whereas several successful so-

lutions have been proposed for the speech synthesis task (Mehri et al., 2016; Oord et al., 2016; Prenger et al., 2019; Wang et al., 2019; Kaneko et al., 2019), the second one is characterized by a key bottleneck: modeling raw audio directly introduces extremely long-range dependencies, making it computationally challenging to learn the high-level semantics of music (Dhariwal et al., 2020).

Nevertheless, only raw audio representation can produce, at least in the long run, appealing results in view of music production for artistic and commercial purposes. Some of the most relevant approaches proposed so far in the field of music generation deal with raw audio representation in the time domain (Oord et al., 2016; Mehri et al., 2016; Dhariwal et al., 2020; Bhave et al., 2019); nevertheless, due to the computational resources required to directly model long-range dependencies in the time domain, either short samples of music can be generated or complex and large architectures and long inference time are required. On the other hand, in (Vasquez & Lewis, 2019) a novel approach is discussed, which proves that long-range dependencies can be more tractably modelled in two-dimensional time-frequency representations such as Mel-spectrograms.

Our work is precisely founded on this novel assumption, thus taking the best from the raw audio representation, while tackling the main issues induced by musical signals long-range dependencies thanks to the waveform-to-spectrograms conversion.

Generative adversarial networks for music generation. Such two-dimensional representation of music paves the way to the application of several image processing techniques and image-to-image translation networks to carry out style transfer and arrangement generation (Isola et al., 2017; Zhu et al., 2017). It is worth recalling that the application of GANs to music generation tasks is not new: in (Brunner et al., 2018), Generative Adversarial Networks are applied on symbolic music to perform music genre transfer; however, to the best of our knowledge, GANs have never been applied to raw audio in the frequency domain for music generation purposes. As to the arrangement generation task, also in this case the large majority of approaches proposed in literature is based on symbolic representation of music: in (Ren et al., 2020), a novel Multi-track MIDI representation (MuMIDI) is presented, which enables simultaneous multi-track generation in a single sequence and explicitly models the dependency of the notes from different tracks by means of a Transformer-based architecture; in (Jiang et al., 2020), a deep reinforcement learning algorithm for online accompaniment generation is described.

Automatic music generation challenges. Coming to the most relevant issues in the development of music generation systems, both the training and evaluation of such systems haven proven challenging, mainly because of the following reasons: (i) the available data sets for music generation tasks are challenging due to their inherent high-entropy (Dieleman et al., 2018), and (ii) the definition of an objective metric and loss is a common problem to generative models such as GANs: at now, generative models in the music domain are evaluated based on the subjective response of a pool of listeners, and just for the MIDI representation a set of simple musically informed objective metrics was proposed (Yang & Lerch, 2020).

3 METHOD

3.1 SOURCE SEPARATION FOR MUSIC

We present a novel framework for automatic music arrangement generation using an adversarially trained deep learning model. A key challenge to our approach is the scarce availability of music data featuring source separated channels (i.e. differentiated vocals, bass, drums, ...). To this end, we leverage Demucs by Défossez et al., a freely available tool, which separates music into its generating sources. This solves the challenge of data availability and allows us to feed our model with the appropriate signals.

It is worth noticing that results of this procedure are time equivariant, meaning that shifting the input mixture by X samples will shift the output Y by the exact same amount. While showing nice properties, at times this method produces noisy separations - with watered–down harmonics and traces of other instruments in the vocal segment - effectively hindering later part of the pipeline.

3.2 MUSIC REPRESENTATION – FROM RAW AUDIO TO MEL-SPECTROGRAMS

One of the main features of our method is to choose a two-dimensional time-frequency representation of the audio samples rather than a time representation. The spectrum is a common transformed representation for audio, obtained via a Fourier transform. Figure 1 shows a Mel-spectrogram example, a visual representation of a spectrum, where the x axis represents time, the y axis represents the Mel bins of frequencies and the third gray tone axis represents the intensity of the sound measured in decibel (Briot et al., 2020). This decision allows to better deal with long-range dependencies typical of such kind of data and to reduce the computational resources and inference time required. Moreover, the Mel scale is based on a mapping between actual frequency and perceived pitch as the human auditory system does not perceive pitch in a linear manner. Finally, using Mel spectrograms of pre-existing songs to train our model potentially enables to draw sounds for new arrangements from the vast collection of music recordings accumulated in the last century. Mel-frequency cepstral coefficients are the dominant features used in speech recognition, as well as in some music modeling tasks (Logan & Robinson, 2001).

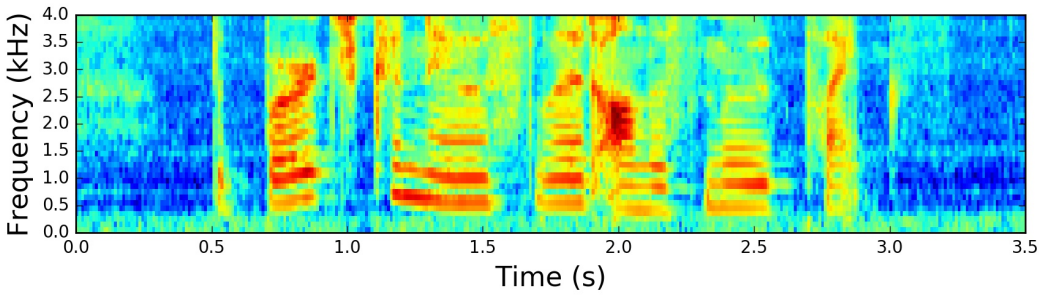

Figure 1: Example of a Mel-spectrogram

After the source separation task was carried out on our song dataset, each source (and the full song) waveforms were turned into corresponding Mel-spectrograms. For the waveform-spectrogram conversion, the sampling rate **s_r** was initially set to 22050 Hz, the number of sampling points to calculate the discrete Fourier transform **n_fft** to 2048, the number of Mel frequency bins **n_mels** to 256 and the step or stride between windows **hop_length** to 512. Finally, we cropped out 256×256 windows from each Mel-spectrograms with an overlapping of 50 time frames, obtaining multiple samples from each song (each equivalent to 5 seconds of music).

3.3 IMAGE TO IMAGE TRANSLATION - CYCLEGAN

The automatic arrangement generation task was faced through an *unpaired image-to-image translation* framework, by adapting the CycleGAN model[1] to our purpose.

CycleGAN is a framework able to translate between domains without paired input-output examples, by assuming some underlying relationship between the domains and trying to learn that relationship. Based on a set of images in domain X and a different set in domain Y, the algorithm learns both a mapping $G : X \rightarrow Y$ and a mapping $F : Y \rightarrow X$, such that the output $\hat{y} = G(x)$ for every $x \in X$, is indistinguishable from images $y \in Y$ and $\hat{x} = G(y)$ for every $y \in Y$, is indistinguishable from images $x \in X$.

The other relevant assumption is that, given a mapping $G : X \rightarrow Y$ and another mapping $F : Y \rightarrow X$, then G and F should be inverses of each other, and both mappings should be bijections. This assumption is implemented by training both the mapping G and F simultaneously, and adding a cycle consistency loss that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$. The cycle consistency loss is then combined with the adversarial losses on domains X and Y (Zhu et al., 2017).

[1]Model available at `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`

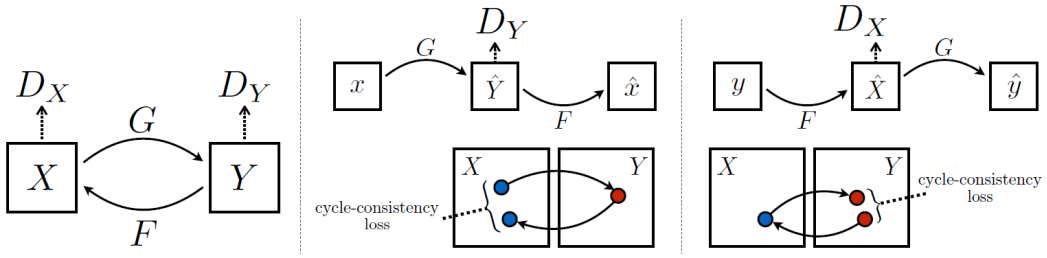

Figure 2: Representation of the CycleGAN model, which consists of two mapping functions G and F, two discriminators D_X and D_Y and two cycle-consistency losses (Zhu et al., 2017)

3.4 Automatic Music Production

The method we propose takes as input a set of N music songs in the waveform domain $X = \{\mathbf{x_i}\}_{i=1}^{N}$, where $\mathbf{x_i}$ is a waveform whose number of samples depends on the sampling rate and the audio length. Each waveform is then separated by Demucs into three different sources. Thus, we end up having four different WAV files for each song, which means a new set of data of the kind: $X_{\text{NEW}} = \{\mathbf{x_i}, \mathbf{v_i}, \mathbf{d_i}, \mathbf{b_i}\}_{i=1}^{N}$, where $\mathbf{v_i}, \mathbf{b_i}, \mathbf{d_i}$ represents vocal, bass, and drums respectively.

Each track is then converted to its Mel-spectrogram representation. Since the CycleGAN model takes 256×256 images as input, each spectrogram is chunked into smaller pieces with an overlapping window of 50 time frames; finally, in order to obtain grayscale PNG images from the original spectrograms, we perform a discretization step in the range $[0 - 255]$.

In the final stage of our pipeline, we feed the obtained dataset to the CycleGan model, that has been adapted to the structure of this data. Even though the discretization step introduces some distortion – original spectrogram values are floats – the impact on the audio quality is negligible.

At training time, we considered two different experimental settings. On the one hand, we take the vocals and the whole song respectively - as the model takes into account two domains X, Y - with the goal of generating an arrangement euphonic to the vocal line. On the other hand, in the second experimental setting we feed the model with bass and drums lines in order to create suitable drums given a bass line.

For both tasks, we trained our model on 2 NVIDIA Tesla V100 SXM2 with 32 GB of RAM for 12 epochs (FMA dataset), and fine-tuned it for 20 more epochs (musdb18 dataset). Each task required 6 days of training time.

For both settings, as a final step, the inverse procedure is applied, to convert back the spectrograms obtained to the waveform domain and to evaluate the produced music.

4 Experiments

4.1 Dataset

For the quality of the generated music samples, it is important to carefully pick the training dataset. To train and test our model We decided to use the Free Music Archive (FMA), and the musdb18 dataset (MusDB18) that were both made available quite recently.

The Free Music Archive (FMA) is the largest publicly available data set suitable for music information retrieval tasks (Defferrard et al., 2016). In its full form it provides 917 GB and 343 days of Creative Commons-licensed audio from 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 unbalanced genres. It provides full-length and high-quality audio, pre-computed features, together with track- and user-level metadata, tags, and free-form text such as biographies. Given the size of FMA, we chose to select only pop music and its sub-genres, for a total of approximately 10,000 songs.

Finally, in order to better validate and fine–tune our model we decided to also use the musdb18 dataset. This rather small dataset is a unique and precious source of songs delivered in multi-track fashion. Each song comes as 5 audio files – vocals, bass, drums, others, full song – perfectly separated at the master level.

In the training set each song is represented by up to 50 samples, depending on its length. In the test set, instead, we chose only a few samples for each song due to the relative uniformity of its content: in other word, we expect our model to perform in similar ways on different parts of the same song. No single song is present in both datasets.

4.2 Experimental setting

There is an intrinsic difficulty in objectively evaluating artistic artifacts such as music. Many generative approaches to raw audio, such as Jukebox (Dhariwal et al., 2020; Mor et al., 2018), try to overcome this obstacle by having the results manually tagged by human experts. Although this rating may be the best in terms of quality, the result is somehow subjective, thus different people may end up giving different or biased ratings based on their personal taste. Moreover, the computational cost and time required to manually annotate the dataset could become prohibitive even for relatively few samples (over 1000). Accordingly, as part of our contribution and taking inspiration from the song intelligibility score (Sharma & Wang, 2019), we propose a new metric that highly correlate with human judgment. This could represent a first benchmark for the tasks at hand.

4.3 Metrics

If we consider as a general objective for a system the capacity to assist composers and musicians, rather than to autonomously generate music, we should also consider as an evaluation criteria the satisfaction of the composer (notably, if the assistance of the computer allowed him to compose and create music that he may consider not having been possible otherwise), rather than the satisfaction of the auditors (who remain too often guided by some conformance to a current musical trend) (Briot et al., 2020).

However, as previously stated, an exclusive human evaluation may be unsustainable in terms of computational cost and time required. Thus we carried out the following quantitative assessment of our model. We first produced 400 test samples – from as many different songs and authors – of artificial arrangements and drum lines starting from voice and bass lines that were not part of the training set. We then asked two professional musicians and two music producers with more than 4 years of experience to manually annotate these samples, capturing the following musical aspects: quality, euphony, coherence, intelligibility. More precisely, for each sample, we asked them to rate from 1 to 10 the following aspects: (i) *Quality*: a rating from 1 to 10 of the naturalness and absence of artifacts or noise, (ii) *Contamination*: a rating from 1 to 10 of the contamination by other sources, (iii) *Credibility*: a rating from 1 to 10 of the credibility of the sample, (iv) *Time*: a rating from 1 to 10 of whether the produced drums and arrangements are on time the the bass and voice lines.

Ideally, we want to produce some quantitative measure whose outputs – when applied to generated samples – highly correlates (i.e. predict) expert average grades. To achieve this goal, we trained a logistic regression model with features obtained through a comparison between the original arrangement and the model output, as well as the original drums and the artificial drums. Here are the details on how with obtained suitable features:

STOI-like features. We created a procedure – inspired by the STOI (Andersen et al., 2017) – whose output vector somehow measures the Mel frequency bins correlation throughout time between the original sample (arrangement/drums) and the fake one. The obtained vector can then be used to feed a multi regression model whose independent variable is the human score attributed to that sample. Here is the formalisation:

$$HumanScore = \sum_{i}^{256} a_i \left[\sum_{t}^{256} (x_i^{(t)} - \bar{x}^{(t)})(y_i^{(t)} - \bar{y}^{(t)}) \right]$$

To simplify, to each pair of samples (original and generated one) a 256 element long vector is associated as follows:

$$\mathcal{S}(\mathcal{X}, \mathcal{Y}, l)^{(i)} = \sum_{t}^{256} (x_i^{(t)} - \bar{x}^{(t)})(y_i^{(t)} - \bar{y}^{(t)})$$

Where:

- $\mathcal{X}$ and $\mathcal{Y}$ are, respectively, the Mel-spectrogram matrices of original and generated samples;
- a_i is the i-th coefficient for the linear regression;
- $x_i^{(t)}$ and $y_i^{(t)}$ the i-th element of the t-th column of matrices $\mathcal{X}$ and $\mathcal{Y}$, respectively;
- $\bar{x}^{(t)}$ and $\bar{y}^{(t)}$ are the means along the t-th column of matrices $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Each feature i of the regression model is a sort of Pearson correlation coefficient between row i of $\mathcal{X}$ and row i of $\mathcal{Y}$ throughout time.

FID-based features. In the context of GANs result evaluation, the Fréchet Inception distance (FID) is supposed to improve on the Inception Score by actually comparing the statistics of generated samples to real samples (Salimans et al., 2016; Heusel et al., 2017). In other words, FID measures the probabilistic distance between two multivariate Gaussians, where $X_r = N(\mu_r, \Sigma_r)$ and $X_g = N(\mu_g, \Sigma_g)$ are the 2048-dimensional activations of the Inception-v3 pool3 layer – for real and generated samples respectively – modeled as normal distributions. The similarity between the two distributions is measured as follow:

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

Nevertheless, since we want to assign a score to each sample, we just estimated the $X_r = N(\mu_r, \Sigma_r)$ parameters – using different activation layers of the Inception pre-trained network – and then we calculated the probability density associated to each fake sample. Finally, we added these scores to the regression model predictors.

4.4 Experimental Results

For the bass2drums task, figure 3 shows the distribution of grades for the 400 test samples - averaged among all four independent evaluators and rounded to the closest integer. The higher the grade, the better the sample will sound. More precisely, after discussing the model results with the evaluators, we noticed that samples with grade 1-3 are generally silent or very noisy. In samples graded 4-5 few sounds start to emerge, but they are usually not very pleasant to listen to, nor coherent. Grades 6-7 identify drums that sound good, that are coherent, but that are not continuous: they tend to follow the bass line too closely. Finally, samples graded 8 and 9 are almost indistinguishable from real drums, both in terms of sound and timing.

In the labeling of non graded samples phase, we therefore assigned a 0 to those samples whose average grade was between 1 and 5, and 1 to those between 6 and 10. Finally, we trained a multi-logistic regression model with both the STOI-like and the FID-based features. The model accuracy on test set was 87%.

Given this pretty good result, we could then used this trained logistic model to label 14000 different 5s fake drums clips, produced from as many real bass lines. Two third of these were labeled as good sounding and on time. Here is a private Sound Cloud playlist where you can listen to some of the most interesting results: `https://soundcloud.com/user-639025674/sets/bass2drums/s-jjccrgdXXOi`.

Regarding instead the voice2song task, results were far less encouraging. Even though some nice arrangements were produced, the model failed to properly and euphonically arrange the input voice lines. For this reason, here we limit to report some of the best produced samples, in the hope to improve the model greatly in the following months: `https://soundcloud.com/user-639025674/sets/voice2song/s-pCPKlQfTbn8`.

As for baselines, initially we thought about comparing our results to three particularly notable works (Dhariwal et al., 2020; Vasquez & Lewis, 2019; Mor et al., 2018), but after running some experiments we eventually realized that they could not be properly used for arrangement purposes. All

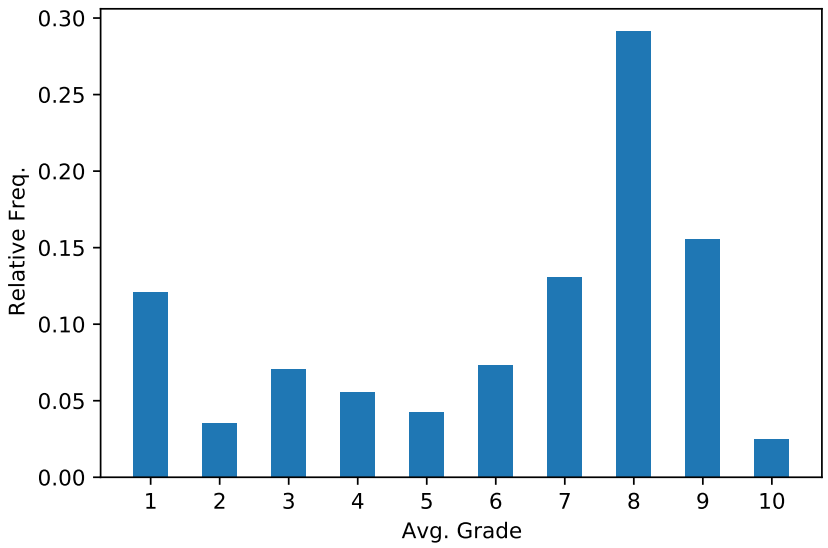

Figure 3: Grade distribution of generated drums samples

three model produce very nice music samples, but none of them can take as input vocals or bass lines and produce a complementary arrangement. It is possible though that these models could be fine tuned to solve this new task.

Finally, with respect to the computational resources and time required to generate new arrangements, our approach shows several advantages, compared to auto-regressive models (Dhariwal et al., 2020), by leveraging the two-dimensional time-frequency representation: since the output prediction can be parallelised, the inference time amounts to few seconds, whereas the Mel-spectrogram-waveform conversion duration depends on the input length, but it never exceeds few minutes. Indeed, it is worth noting that, at inference time, arbitrary long inputs can be processed and arranged.

5 Conclusions and Future Work

In this work, we presented a novel approach to automatically produce euphonic music arrangements starting from a voice line or a bass line. We applied Generative Adversarial Networks to real music pieces, treated as grayscale images (Mel spectrograms). Given the novelty of the problem, we proposed a reasonable procedure to properly evaluate our model outputs. Notwithstanding the promising results, some critical issues need to be addressed before a more compelling architecture can be developed. First and foremost, a larger and cleaner dataset of source separated songs should be created. In fact, manually separated track always contain a big deal of noise. Moreover, the model architecture should be further improve to focus on longer dependencies and to take into account the actual degradation of high frequencies. Finally, a certain degree of interaction and randomness should be inserted to make the model less deterministic and to give creators some control over the sample generation. Our contribution is nonetheless a first step toward more realistic and useful automatic music arrangement systems and we believe that further significant steps could be made to reach the final goal of human-level automatic music arrangement production. Already now software like Melodyne (Neubäcker, 2011; Senior, 2009) delivers producers a powerful user interface to directly intervene on a spectrogram-based representation of audio signals to correct, perfect, reshape and restructure vocals, samples and recordings of all kinds. In is not unlikely that in the future artists and composers will start creating their music almost like they were drawing.

References

Asger Heidemann Andersen, Jan Mark de Haan, Zheng-Hua Tan, and Jesper Jensen. A non-intrusive short-time objective intelligibility measure. In *2017 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), pp. 5085–5089. IEEE, 2017.

Gérard Assayag, Camilo Rueda, Mikael Laurson, Carlos Agon, and Olivier Delerue. Computer-assisted composition at ircam: From patchwork to openmusic. *Computer music journal*, 23(3): 59–72, 1999.

Aishwarya Bhave, Mayank Sharma, and Rekh Ram Janghel. Music generation using deep learning. In *Soft Computing and Signal Processing*, pp. 203–211. Springer, 2019.

Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. *Deep learning techniques for music generation*. Springer, 2020.

Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Sumu Zhao. Symbolic music genre transfer with cyclegan. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 786–793. IEEE, 2018.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Demucs: Deep extractor for music sources with extra unlabeled data remixed. *arXiv preprint arXiv:1909.01174*, 2019.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Sander Dieleman, Aaron van den Oord, and Karen Simonyan. The challenge of realistic music generation: modelling raw audio at scale. In *Advances in Neural Information Processing Systems*, pp. 7989–7999, 2018.

Marko Docevski, Eftim Zdravevski, Petre Lameski, and Andrea Kulakov. Towards music generation with deep learning algorithms, 2018.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *arXiv preprint arXiv:1709.06298*, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Nan Jiang, Sheng Jin, Zhiyao Duan, and Changshui Zhang. Rl-duet: Online music accompaniment generation using deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 710–718, 2020.

Tianyu Jiang, Qinyin Xiao, and Xueyuan Yin. Music generation using bidirectional recurrent network. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pp. 564–569. IEEE, 2019.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6820–6824. IEEE, 2019.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*, pp. 14910–14921, 2019.

Beth Logan and Tony Robinson. Adaptive model-based speech enhancement. *Speech Communication*, 34(4):351–368, 2001.

Rachel Manzelli, Vijay Thakkar, Ali Siahkamari, and Brian Kulis. An end to end model for automatic music generation: Combining deep raw and symbolic audio networks. In *Proceedings of the Musical Metacreation Workshop at 9th International Conference on Computational Creativity, Salamanca, Spain*, 2018.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. *arXiv preprint arXiv:1805.07848*, 2018.

Peter Neubäcker. Sound-object oriented analysis and note-object oriented processing of polyphonic sound recordings, September 20 2011. US Patent 8,022,286.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Manan Oza, Himanshu Vaghela, and Kriti Srivastava. Progressive generative adversarial binary networks for music generation. In *International Conference on Innovative Computing and Communications*, pp. 181–192. Springer, 2020.

Alexandre Papadopoulos, Pierre Roy, and François Pachet. Assisted lead sheet composition using flowcomposer. In *International Conference on Principles and Practice of Constraint Programming*, pp. 769–785. Springer, 2016.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621. IEEE, 2019.

Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Popmag: Pop music accompaniment generation. *arXiv preprint arXiv:2008.07703*, 2020.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

M Senior. Celemony melodyne dna editor. *Sound on Sound*, 2009.

Bidisha Sharma and Ye Wang. Automatic evaluation of song intelligibility using singing adapted stoi and vocal-specific features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:319–331, 2019.

Sean Vasquez and Mike Lewis. Melnet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083*, 2019.

Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415, 2019.

Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 32(9):4773–4784, 2020.

Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*, 2017.

Yi Zhao, Xin Wang, Lauri Juvela, and Junichi Yamagishi. Transferring neural speech waveform synthesizers to musical instrument sounds generation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6269–6273. IEEE, 2020.

Hongyuan Zhu, Qi Liu, Nicholas Jing Yuan, Chuan Qin, Jiawei Li, Kun Zhang, Guang Zhou, Furu Wei, Yuanchun Xu, and Enhong Chen. Xiaoice band: A melody and arrangement generation framework for pop music. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2837–2846, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.