Convolutional Generative Adversarial Network, via Transfer Learning, for Traditional Scottish Music Generation

Francesco Marchetti[1][0000−0003−4552−0467], Callum Wilson[1][0000−0003−3736−1355], Cheyenne Powell[1][0000−0001−9343−0664], Edmondo Minisci[1][0000−0001−9951−8528], and Annalisa Riccardi[1][0000−0001−5305−9450]

[1]Intelligent Computational Engineering Laboratory (ICE Lab)
Department of Mechanical and Aerospace Engineering
University of Strathclyde, Glasgow, UK
{name.surname}@strath.ac.uk

Abstract. The concept of a Binary Multi-track Sequential Generative Adversarial Network (BinaryMuseGAN) used for the generation of music has been applied and tested for various types of music. However, the concept is yet to be tested on more specific genres of music such as traditional Scottish music, for which extensive collections are not readily available. Hence exploring the capabilities of a Transfer Learning (TL) approach on these types of music is an interesting challenge for the methodology. The curated set of MIDI Scottish melodies was preprocessed in order to obtain the same number of tracks used in the BinaryMuseGAN model; converted into pianoroll format and then used as training set to fine tune a pretrained model, generated from the Lakh MIDI dataset. The results obtained have been compared with the results obtained by training the same GAN model from scratch on the sole Scottish music dataset. Results are presented in terms of variation and average performances achieved at different epochs for five performance metrics, three adopted from the Lakh dataset (qualified note rate, polyphonicity, tonal distance) and two custom defined to highlight Scottish music characteristics (dotted rhythm and pentatonic note). From these results, the TL method shows to be more effective, with lower number of epochs, to converge stably and closely to the original dataset reference metrics values.

Keywords: Generative Adversarial Network · Transfer learning · Convolutional Neural Network · Scottish Music.

1 Introduction

The ability for Artificial Intelligence (AI) to generate music is a challenge that has been taken on by many for various reasons. Music is known to assist humans with emotional comfort and needs and thus the ability for a machine to create such melodies to accomplish that is being looked into with the advancement of technology [18].

The first examples dated in 1957 and 1958 are known as the Illiac Suite, which used a process of music composition through sequential record of experiments performed by a computer [21]. The next occurred in 1960, when R. Kh. Zaripov explored the use of a computer for creating short monophonic melodies, with Zaripov later identifying the creation of melodies as being one of the most important aspects of the process [1].

Progress continued throughout the years of 1975 and 1980, as Mark Steedman investigated machine perception of musical rhythms [22] and David Cope's experiments in musical intelligence focused on imitating his own musical style [6]. This paved the way for major companies like Google and Sony to begin development in their own laboratories which have produced a number of computer-generated music compositions. The Google Brain team, Magenta; took two approaches to understand the progress of audio modelling. The first used the WaveNet style auto-encoder that is used on temporal codes by the conditioning of an auto-regressive decoder to learn from the raw audio waveforms. The second was the introduction of a data set consisting of musical notes in the order of a larger scale of similar public data-sets known as NSynth [9]. Sony's technologies on the other hand were used in various ways to create different types of music. The first of these is titled "Daddy's Car", which was created with the inspiration of the Beatles and another by SKYGGE to generate an album that consisted of an AI-human collaboration titled "Hello World" [2].

Generative Adversarial Networks (GANs) are one of the methods used for AI music generation. A GAN is a framework proposed by Goodfellow et. al. [11] where two Neural Networks (NNs) are trained simultaneously in an adversarial manner. One network is called the *Generator* and the other is called the *Discriminator*. The Generator creates samples starting from uniformly distributed random data and the Discriminator receives as input either the sample generated by the Generator network or real data. The goal of the Discriminator is to learn weather the data that it receives as input are samples generated by the Generator or are real data. The goal of the Generator is to learn how to "trick" the Discriminator into making it believe that the samples generated by the Generator are real data. In this way, the Generator learns how to produce samples which resemble real data but are completely artificial [14].

GANs are used for several tasks, resulting in different types of GAN specific to each task. A few of these are: cGAN, a type of conditional GAN used to aid computer diagnosis systems in the localisation and detection of prostate tissue on MRI scans [12]. CycleGAN, an image-to-image translation approach where image mapping is learned using image pairs that are aligned [25], in addition to being used for images they have also been applied to the concept of music in the form of a symbolic music representation in MIDI format [4]. FusionGAN, also used for generating music, is a type of fusion framework and an optional use of dual learning that can be implemented on the styles of the provided domains [5].There is also a style-based GAN architecture (StyleGAN) for image modelling [15] and an Image Super-Resolution GAN (SRGAN) which uses a low-resolution image to estimate a corresponding high-resolution image [16]. Overall, various types of GANs are used for art generation, whether it is a picture, a video, or music. Some examples of their art generation capabilities involve the generation of photographs of human faces [14], the generation of pictures from a text description [24], the prediction of video frames up to a second [23], and music generation [17].

A Binary Multi-track Sequential GAN (BinaryMuseGAN) is used for generating multi-track polyphonic music consisting of multi-track inter-dependency, temporal, and harmonic and rhythmic structures. Two different scenarios were taken to integrate the temporal model; one of these scenarios incorporates the learning of a temporal structure from a human made track while the other is the ability to make its own music without any human intervention. With the use of these principles, private Generators are used for individual tracks; another used with a combination of private generators for each track and their inputs that are shared among tracks; and last is another approach with all tracks executed at once with one Generator. When results are combined to view bars, Convolutional Neural Networks (CNNs) are used to translate patterns [7]. In this paper, BinaryMuseGANs will be used throughout for the unsupervised generation of Scottish traditional music patterns.

The paper is divided in three more sections, excluding introduction, to highlight the main contribution of the paper

- In Section 2 the curated Scottish music dataset is presented together with the proposed instruments mapping used to associate Scottish traditional instruments into respective BinaryMuseGAN tracks list. Data curation and preparation of Scottish traditional music for evolutionary approaches has never been completed or studied before, hence it represents the first contribution of our work.
- In Section 3 the training methodology used is presented. The main contribution of our paper is the utilisation of a TL scheme where a pre-learned model, trained with BinaryMuseGAN on a dataset of non specialist music pianoroll tracks (Lakh MIDI dataset), is fine tuned on the Scottish music dataset. This has a double advantage: the model can be fine tuned effectively on smaller datasets, the training can achieve greater performances because the core features of music generation has already been learnt in the pre-learning phase. TL approaches for music generation is a largely unexplored area, to the best of our knowledge no previous approach has attempted anything similar.
  - In Subsection 3.1 two new metrics, dotted rhythm and pentatonic note, are defined, as deemed to be representatives of the key features of Scottish music.
- Finally in Section 4, the training of the TL model is compared with the model trained from scratch. The results show the superiority of a TL approach in terms of robustness and metrics performances.

2 Data gathering and preprocessing

The training data consists of 137 midi files sequenced by Barry Taylor which are traditional or contemporary Scottish tunes [3]. Here the process used to clean and preprocess the data is described. This is a non-trivial task as the data have many differing characteristics. Starting with the time signatures, these are used to narrow down the dataset to suitable files. Tracks in the raw dataset are then assigned to new tracks in the processed dataset depending on their instrumentation and certain other features as will be described. Finally, the shaping of the data in a format which can be used to train the GAN is described.

2.1 Time signatures

First the time signature of each file is considered, which is stated in the midi file metadata. In the original BinaryMuseGAN dataset [8], the authors only use pieces with a time signature of 4/4. Traditional Scottish music often features compound times such as 6/8 or 9/8, which would therefore be unsuitable to allow TL. The proportions of the time signatures in the Scottish music dataset are shown in Figure 1. Since 4/4 time signatures only make up 69.2% of the total dataset, it was decided also to include 2/4 and 2/2 as these have a very similar feel to 4/4. This reduces the valid dataset size to 78 files.

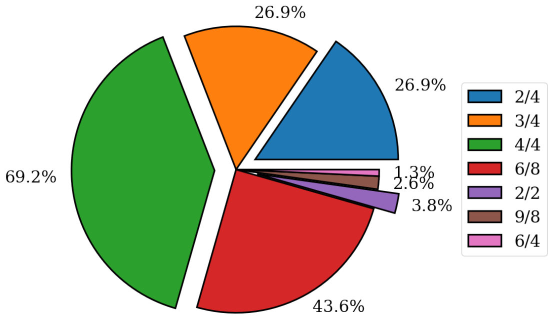


Fig. 1: Proportion of each time signature present in the Scottish music dataset - exploded segments indicate the time signatures included in the processed dataset

2.2 Tracks and instrumentation

Now the instrumentation of each file is considered. Each track in a file has an associated General MIDI program change number - equivalent to an instrument [20]. BinaryMuseGAN generates 8 tracks with the following midi instruments: Drums, Piano, Guitar, Bass, Ensemble, Reed, Synth Lead, and Synth Pad. Effective TL requires the same number of tracks to be generated, however the exact instrumentation used in their implementation is not suitable for a Scottish ensemble. In our dataset, certain files contain multiple tracks of the same instrument; for example, a file with 4 tracks could have all tracks with program change number 1, which is an Acoustic Grand Piano. Most files have fewer than 8 tracks which is highlighted in Figure 2. As a result, the preprocessed dataset is sparse with respect to which of the 8 tracks are not empty. The instrumentation was adjusted to the following: Drums, Piano, Guitar, Bass, Fiddle, Wind, Accordion, and Clarsach. These are listed in Table 1 along with their corresponding General MIDI program change number. The table also shows which program change numbers (i.e. instruments) from the raw dataset are included in each new track.

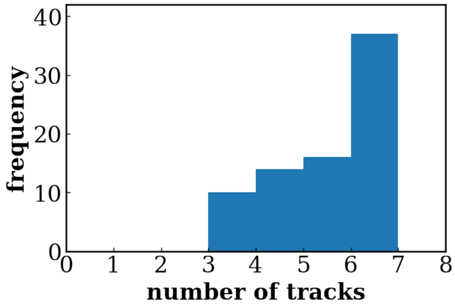


Fig. 2: Histogram of the number of tracks in each file in the Scottish music dataset.

Table 1: Instrument changes for scottish music generation

Track no.	Their instrument	Our instrument	Included instruments
0	Drums	Drums	Drums
1	Piano	Piano (Acoustic Grand Piano - 1)	1,2,3,5,6,8
2	Guitar	Guitar (Acoustic Guitar (steel) - 26)	25,26,31,106
3	Bass	Bass (Electric Bass (finger) - 34)	33,34,36,37,43
4	Ensemble	Fiddle (111)	41,42,49,51,111
5	Reed	Wind (Recorder - 75)	57,58,59,62,66,67, 2,68,69,71,72,73, 74,75,76,77,80,110,11
6	Synth Lead	Accordion (22)	22,23,24
7	Synth Pad	Clarsach (Orchestral Harp - 47)	11,15,16,89,95,47

We refer to the 8 tracks in the preprocessed dataset as “new tracks” and tracks from files in the raw dataset as “old tracks”. Any new track for a given file which contains one or more old tracks is referred to as a non-empty track, whereas empty tracks do not contain any old tracks. Only using the transformations shown in Table 1 results in very uneven distributions of old tracks among new tracks which leaves many new tracks empty as shown in Figure 4. To avoid having large numbers of new tracks in the dataset which do not contain any notes, the old tracks should be spread more evenly across the new tracks. To achieve this balancing, heuristics are employed to change some of the tracks based on the following features of each track: mean note, number of notes, and polyphonic ratio.

Every note in any track is represented by a number from 0 to 127 which defines the note's pitch as per the General MIDI standard [20]. The processed format of the dataset is a pianoroll, which is an array with a temporal dimension and pitch dimension. Along the temporal dimension, every column contains 128 binary-valued pitches which indicate whether that pitch is being played at that time or not. This requires the definition of a frequency parameter that controls the effective resolution of the temporal dimension. The same frequency of 24 as the original MuseGAN research was used, which means every quarter-note beat has 24 timesteps and every bar of four beats contains 96 timesteps.

To calculate the mean note, number of notes (here denoted n_{note}), and polyphonic ratio (herein referred to as "poly ratio") for a given track it is first converted to the pianoroll format. The number of notes is the number of nonzero locations in the pianoroll. While this does not take into account notes which are sustained for multiple timesteps, it can still give an indication of how many notes are played in that track. The mean note is the average location of nonzero elements along the pitch dimension. Finally, the poly ratio is the ratio between the number of locations along the temporal dimension with more than one nonzero location to the number of locations along the temporal dimension with any nonzero location. This indicates whether the track is playing mostly chords of multiple notes or mostly single lines of notes.

To balance the tracks, we derived heuristics based on these metrics for moving old tracks from non-empty tracks which contain three or more old tracks to other, "target" new tracks. These heuristics are designed to move old tracks to target new tracks with suitable instrumentation, e.g., tracks which play low notes go to bass, tracks which play more chords go to piano. The heuristics used are summarised in Algorithm 1.

Algorithm 1 Psuedocode of the heuristic used to balance the tracks

```
if mean note $< 50$ then
    move to bass
end if
if mean note $> 60$ and $n_{note} > 100$ then
    move to fiddle or wind
end if
if poly ratio $> 0$ then
    move to piano
end if
if poly ratio $< 0.1$ then
    move to any empty track except drum or bass
end if
```

The final heuristic in Algorithm 1 (lines 10-12) is used in case not enough suitable tracks are found using the other rules. The decision to move an old track to a different new track using the heuristics depends on the number of old tracks in the current new track and sometimes on the number of old tracks already in the target new track, as described here. Let n_{ne} denote the number of non-empty new tracks, n^i_{new} denote the number of old tracks in new track i, and n^t_{new} denote the number of old tracks in a target track t to which an old track could be moved. To balance the new tracks, the aim is to have all $n^i_{new} <= 2$ and $n_{ne} >= 3$. First new tracks with $n^i_{new} > 2$ are found and the rules above are applied to those tracks until all $n^i_{new} <= 2$. Then if n_{ne} is still less than 3, the tracks where $n^i_{new} = 2$ are considered and the rules are applied again. This is shown schematically in Figure 3 which describes which tracks are moved and the conditions necessary for moving.

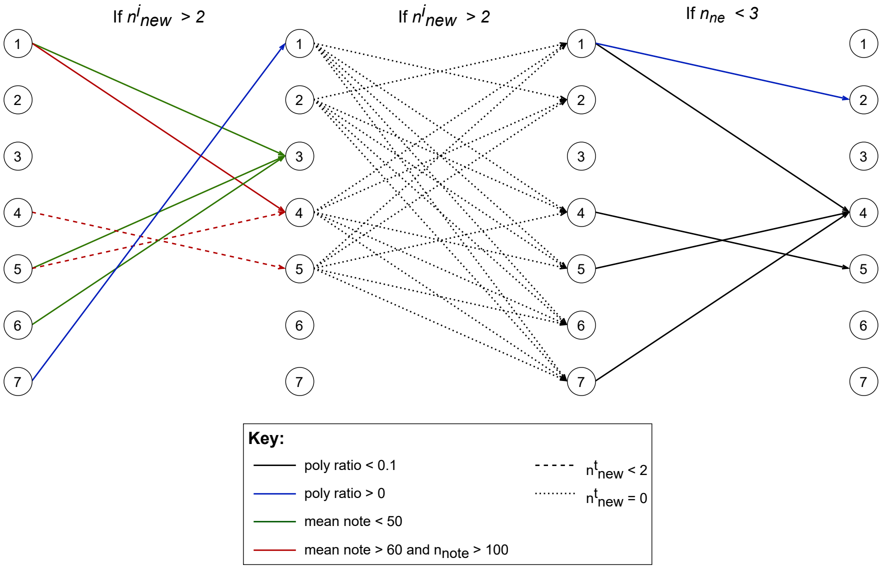
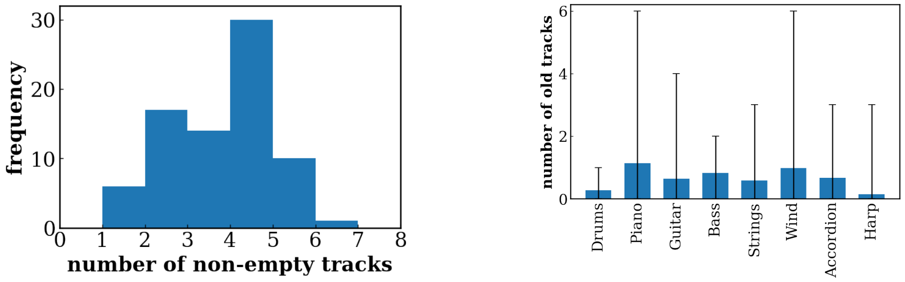


Fig. 3: Graph showing the heuristics used to balance the new tracks. Line colour indicates the condition applied to the old track to be moved. Dashed or dotted lines indicate the condition on the number of old tracks in the target track. Track 0 is drums and is not shown here since no tracks are transferred to or from this track.

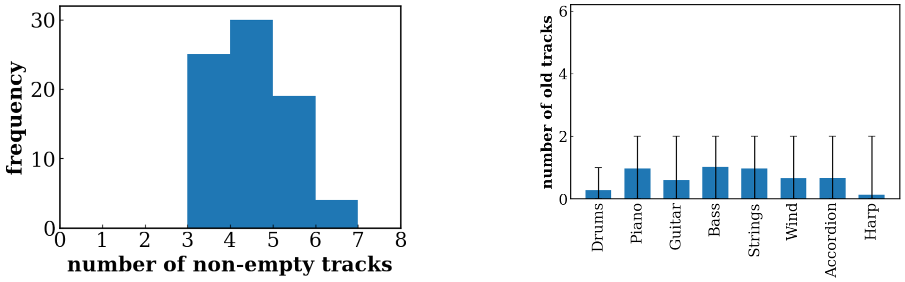
Figure 5 shows the number of non-empty tracks and the distribution of old tracks after applying these balancing heuristics. Compared to Figure 4, the number of non-empty tracks for each file is greater on average with none having fewer than 3 non-empty tracks. Moreover, considering the number of old tracks in each new track, the maximum is now 2 for all new tracks except drums and the average is more evenly distributed across the new tracks.



(a) Histogram showing numbers of new tracks for each file in the Scottish music dataset

(b) Bar chart of average number of old tracks in each new track - errorbars indicate minimum and maximum

Fig. 4: Balance of tracks in the dataset after applying track conversions from Table 1.



(a) Histogram showing numbers of new tracks for each file in the Scottish music dataset

(b) Bar chart of average number of old tracks in each new track - errorbars indicate minimum and maximum

Fig. 5: Balance of tracks in the dataset after applying track conversions from Table 1 and further track balancing from Figure 3.

2.3 Data shaping

The final step in preprocessing the data is shaping the pianoroll array. As discussed previously, the pianoroll format has a temporal and pitch dimension. This new dataset should also be the same shape as the original BinaryMuseGAN dataset to effectively apply TL. In the original work, the array is cropped in the pitch dimension to 84 notes between 24 (C1) and 107 (B7) and the same was done here. Since none of the notes in the Scottish music dataset are outwith this range, this does not cause any notes to be lost.

Along the temporal dimension, data are split into bars of 96 timesteps. Each point in the dataset contains 4 bars. The final dimensions of the array are then [*number of data points*×*4(bars)*×*96(timesteps)*×*84(note pitches)*×*8(tracks)*]. For the original BinaryMuseGAN dataset, each piece has 6 4 bar phrases randomly sampled and added to the dataset. Due to the limited number of files in the Scottish music dataset, all of the 78 valid pieces are instead divided into 4 bar phrases of 96 timesteps and all of these are included in the dataset. This gives 1047 data points.

3 GAN model

The technique chosen in this work to train a network able to learn from the curated dataset of Scottish music, is an adaptation of the GAN model developed by Dong et. al. [8]. They developed a deep convolutional GAN that employs binary output neurons to generate music in the pianoroll format described in Section 2. The model developed in [8] is depicted in Fig. 6 and is composed by:

- a Generator network shared among all the tracks, G_s in Fig. 6, which is responsible of generating a high-level representation of the output music shared by all the tracks. The shared Generator is composed by an input dense layer with 1536 neurons and five transposed convolutional layers.
- A private Generator network for each track, G_p in Fig. 6, which convert the high-level music output provided by the shared generator into the final pianoroll output for the corresponding track. Each private Generator network is composed by three transposed convolutional layers.
- A Refiner network for each track, which refines the real-valued output of the Generators into binary ones. In this network, the tensor size remains unaltered.
- A private Discriminator for each track, D_p in Fig. 6, which extracts low-level features from the corresponding track. Each private Discriminator network is composed by three convolutional layers.
- A Discriminator network shared among all the tracks, D_s in Fig. 6, which extracts a high-level abstraction. The shared discriminator is composed by two convolutional layers.
- An onset/offset stream Discriminator, D_o in Fig. 6, formed by three convolutional layers.
- An chroma stream Discriminator, D_c in Fig. 6, formed by two convolutional layers.
- A final Discriminator, D_m in Fig. 6, which takes as input the outputs of D_s, D_o and D_c. This last Discriminator is composed by one convolutional layer and two dense layers respectively with 1536 and 1 neurons.

The output of the Generator group (shared plus private ones) has the shape $\mathbb{R}^{4\times 96\times 84\times 8}$, which is the same of the input for the Discriminator group. The output of the Discriminator group has shape $\mathbb{R}^1$. The total number of parameters for the BinaryMuseGAN model is 3735737, 1580440 in the Generator group and 2155297 in the Discriminator one. For more detailed informations on each network topology and on the onset/offset and chroma streams, please see [8].

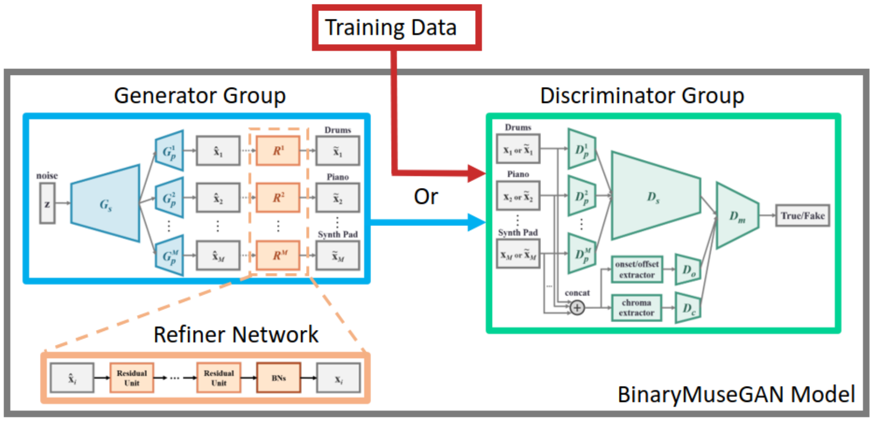


Fig. 6: High level depiction of the BinaryMuseGAN model. The separate images were taken from [8]. This image depicts the high level interaction between the three type of networks. The Generator group receives as input random data and it produces a music sample by first passing through the Refiner network which produces a binary output. The Discriminator group takes as input either training data or the samples produced by the generator and try to determine if the input was either real or fake data.

Such GAN model can be either trained from scratch or via a Transfer Learning framework as shown in Fig. 7 by using a pretrained model. Transfer Learning is a Machine Learning (ML) technique where a model trained on a set of data is used as a starting point for training on a new set of data. It can be used to enhance the generalisation capabilities of the pretrained model, by training it again on different kinds of data; or it can be used to speed up the training process on a set of training data similar to the one used for the pretraining phase; or it can be used to train complex model where large datasets are not available. In this last case, the model is pretrained on a larger alike dataset and then fine tuned on the dataset of interest. In this work the aforementioned GAN model was tested both by training it from scratch on the dataset presented in section 2 and via transfer learning with the pretrained model provided by Dong et. al. [8]. The dataset used to pretrain the model is the piano-roll version of the Lakh dataset, which is a collection of 176,581 unique MIDI files from the Million Song Dataset. This dataset was proposed by Raffel [19] and was converted into piano-roll format by Dong et. al. [7].

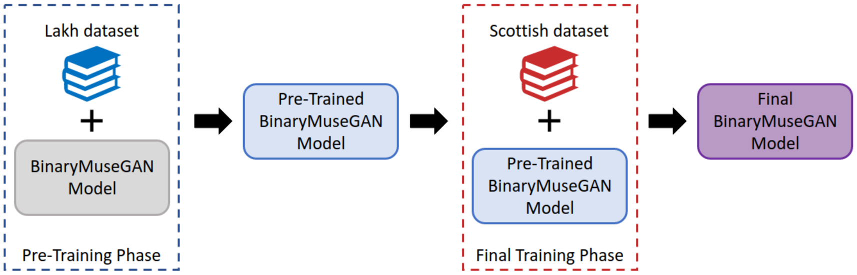

Lakh dataset
+
BinaryMuseGAN Model
Pre-Training Phase
Pre-Trained BinaryMuseGAN Model
Scottish dataset
+
Pre-Trained BinaryMuseGAN Model
Final Training Phase
Final BinaryMuseGAN Model

Fig. 7: Schematic of the Transfer Learning framework employed in this work

3.1 Performance Metrics

To evaluate the results obtained, the same metrics employed in [8] were used, plus two new metrics here which are features typically observed in Scottish music [10], the Dotted Rhythms and Pentatonic Notes. The full set of metrics used are:

- Qualified note rate: evaluates the ratio of the number of qualified notes to the total number of notes. A qualified note is a note no shorter than three time steps. Hence a low qualified note rate value means that the produced track is overly-fragmented.
- Polyphonicity: is the ratio of the number of time steps where more than two pitches are played simultaneously to the total number of time steps.
- Tonal distance: is the distance between the chroma features (one for each beat) of a pair of tracks in the tonal space [13]. As for what was done is [8], also here the tonal distance was measured between the piano and guitar tracks. A larger tonal distance implies weaker inter-track harmonic relations.
- Dotted rhythms: these rhythms feature regularly in Scottish music which are a dotted quaver (18 timesteps) followed by a semiquaver (6 timesteps) or vice versa. In particular, the rhythm of a semiquaver followed by a dotted quaver is often referred to as a “scotch snap” due to its prevalence in traditional Scottish music. This metric assesses what proportion of beats (i.e. 24 timesteps) in a section of a piece contain a dotted rhythm.
- Pentatonic Notes: the pentatonic scale is commonly used in Scottish music, as well as many other styles of music due to its versatility. Since an indication of key signature for the pieces in the Scottish music dataset is not provided, the pentatonic scale to which compare the notes to must be inferred, based on the notes in the piece. To do this, each samples section of a piece was compared to all possible pentatonic scale (starting on each of the 12 possible semitones) and see which gives the highest proportion of pentatonic notes.

The Qualified note rate and the Poliphonicity are defined as intra-track metrics, since they capture the features of the different track separately, while the tonal distance is defined as an inter-track metric since it captures the relation between different tracks. For more informations on these metrics, please see [7,8].

4 Results

As explained in the previous sections, the aim of this work is to use a GAN model to create original music samples from a training database consisting of traditional Scottish music. The used code and the training data can be found at https://github.com/strath-ace/HAGGIS, while the produced results are available from the University of Strathclyde KnowledgeBase at https://doi.org/10.15129/4ae2eb7e-678d-4644-90ad-1cf2a953287f. To assess the effectiveness of this approach, the training process of the model described in Section 3 was repeated a total of 40 times: 20 times where the model was trained from scratch and 20 times where TL was used. Each training simulation was run for 100 epochs to asses its convergence. The obtained results are reported in Table 2 and in Figures 8 and 9.

In Table 2, the values of the metrics of the Scottish dataset are used as reference to quantify the performance of the proposed methodology. The aim is to achieve closer values of the metrics of the results obtained to the reference ones. Besides the reference values, the values of the metrics evaluated on the produced samples at 10, 20 and their average between 20 to 100 epochs are listed. These results are expressed in terms of median and standard deviation, except for the values from 20 to 100 epochs, which represent an average of the median and standard deviation values obtained in the considered epochs range. Data are not available for the dotted rhythm and pentatonic notes metrics prior to 20 epochs because the new metrics were evaluated on MIDI samples output from the training, which by default were produced starting from epoch 20.

Table 2: Median and standard deviation metrics values at 10, 20 and between 20 and 100 Epochs. Scratch refers to the model trained from scratch while TL to the one trained using Transfer Learning. Training refers to the corresponding metric values of the training data. The highlighted values are those with a median value closer to the Training values for each case.

	Training	10 Epochs		20 Epochs		20-100 Epochs	
		Scratch	TL	Scratch	TL	Scratch	TL
Qualified Note Rate	0.987	0.205± 0.091	**0.583±0.055**	0.450± 0.162	**0.583±0.053**	**0.715±0.083**	0.582± 0.040
Poliphonicity	0.393	0.077± 0.061	**0.094±0.019**	0.053± 0.015	**0.097±0.012**	0.077± 0.022	**0.126±0.018**
Tonal Distance	1.394	0.728± 0.283	**1.381±0.156**	**1.374±0.394**	1.334± 0.201	1.335± 0.301	**1.342±0.143**
Dotted Rhythm	0.063	-	-	**0.023± 0.029**	0.115± 0.032	**0.085± 0.050**	0.120± 0.039
Pentatonic Notes	0.594	-	-	**0.542± 0.121**	0.466± 0.066	**0.487± 0.080**	0.437± 0.061

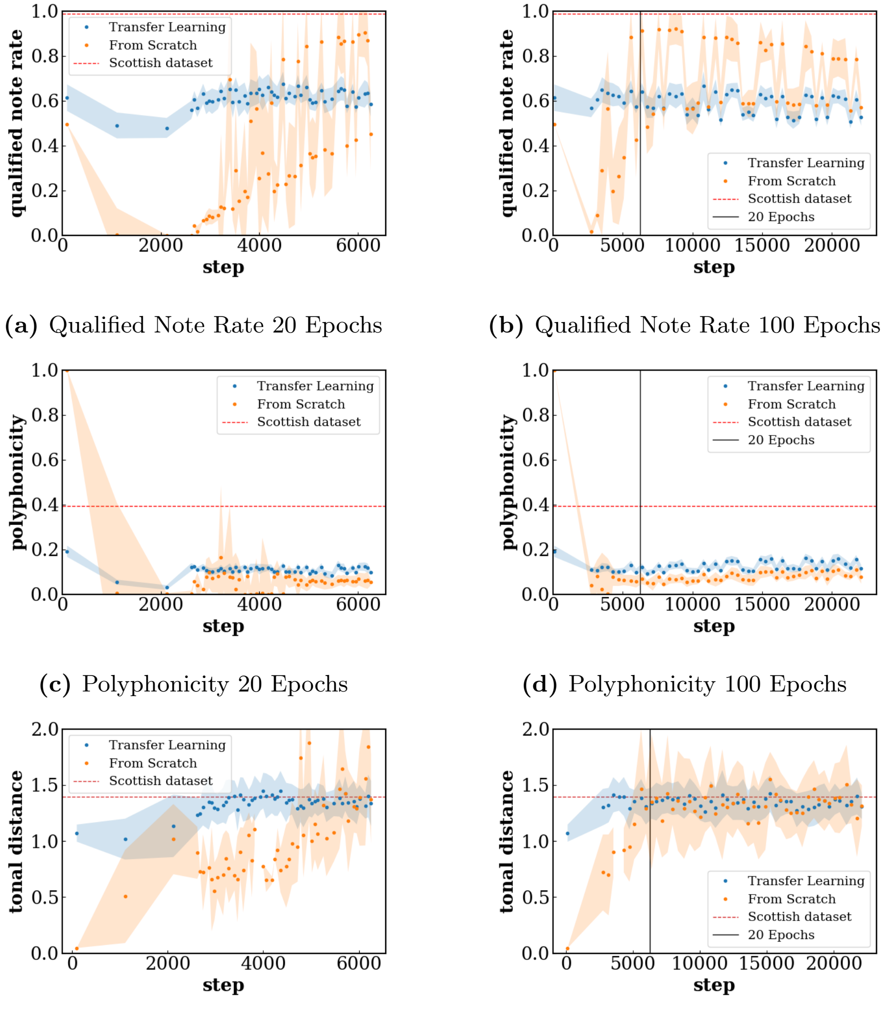
From these results, it can be observed that in the three reported cases (10, 20, 20-100 epochs), the TL approach achieves better results on the majority of the considered conventional metrics in terms of median and standard deviation values. About the two cases that go against this trend, namely the Tonal Distance at 20 Epochs and the Qualified Note Rate at 20-100 Epochs, it can be observed that, in both cases, the model trained from scratch achieves a slightly better median value despite having a standard deviation about twice than the one achieved by the model trained with TL. Regarding the standard deviation, it is clear that for all cases the value measured on the model trained with TL is always smaller than that of the model trained from scratch. Hence more robust results are obtained using the TL approach. This can also be observed by looking at Figures 8a-8f. These plots represent the evolution of the metrics during the training process. In these six figures, the plots on the left are up to 20 epochs to give greater resolution over the initial epochs, while those on the right are up to 100 epochs. It is clear that the use of TL obtains more robust results with respect to these conventional metrics, i.e. with a lower standard deviation, and also with a less oscillating behaviour.

As briefly discussed in Section 3, one motivation for using TL is to obtain superior results with fewer training epochs than the model trained from scratch. This is demonstrated by the results of the TL case after 10 Epochs, which are very close to those obtained at 20 Epochs or to the average computed from 20 to 100 Epochs. On the other hand, the model trained from scratch tends to produce better results as the number of epochs increases.

While the TL approach shows better performance for most metrics, this is not the case for the results of the Scottish metrics. As can be seen from Table 2, at all epochs where data are available, even though the results of TL and learning from scratch are close to each others, the median values of the learning from scratch approach are the ones closer to the reference value of the training data. This is also shown in Figure 9. This suggests that although TL generates most aspects of what can be considered “good” music more quickly, it does not capture the characteristics of a new dataset as well as training from scratch. To obtain values of the Scottish metrics closer to the reference values using the TL strategy, one approach could be integrating these metrics into the formulation of the loss function of the second stage training. This would steer the learning process to the characteristics of Scottish music, while taking full advantage of the robustness and performance recorded for the other metrics.

5 Conclusions

The paper presents an application of the Binary Multi-track Sequential Generative Adversarial Network (BinaryMuseGAN) to generate original Scottish music from a reduced dataset of songs by exploiting a TL approach. The GAN model is first trained on a larger and diverse music collection, to learn representative features of music in general, and then fine tuned on the smaller dataset of traditional Scottish music. The proposed approach demonstrates that more robust and performing results can be obtained via a TL method than by training the same network from scratch with the sole smaller dataset. The results are evaluated with three standard metrics, taken from the literature and two novel metrics defined here to highlight Scottish music characteristics. The results obtained for these last two metrics show that, despite the TL approach achieving superior



(a) Qualified Note Rate 20 Epochs

(b) Qualified Note Rate 100 Epochs

(c) Polyphonicity 20 Epochs

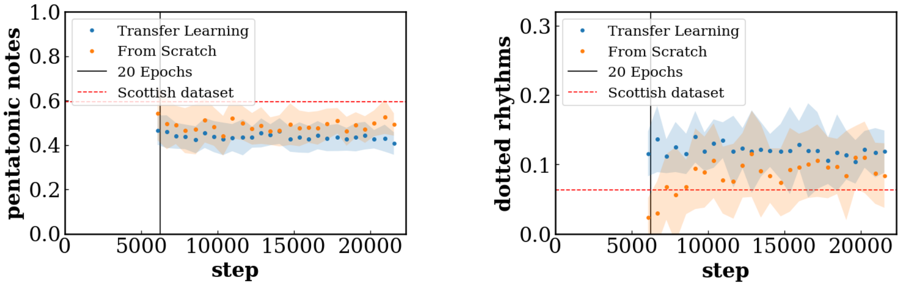
(d) Polyphonicity 100 Epochs

(e) Tonal Distance 20 Epochs

(f) Tonal Distance 100 Epochs

Fig. 8: Qualified Note Rate, Polyphonicity and Tonal Distance evolution during the training process. The plots on the left are up to 20 Epochs, while those on the right are up to 100. On the right plots, the vertical line represents where 20 epochs is.

training performance according to standard metrics, the most relevant features of Scottish music are lost in the process if compared to a learning from scratch approach. This is mainly due to imbalance between the two datasets used and no adaptation of the loss function formulation during the fine tuning training step. While the proposed approach is an initial step in generating original music



(a) Pentatonic notes

(b) Dotted rhythms

Fig. 9: Pentatonic notes and Dotted rhythms metrics evolution during the training process.

from reduced datasets of specific music kind, some limitations are highlighted for future studies. While for the TL approach the BinaryMuseGAN network topology needs to be invariate from the one used in the pretraining phase, for the learning from scratch approach, the topology of the network can be further optimised to improve the results. In addition, unsupervised analysis techniques such as clustering or principal components analysis could provide further insights into differences between datasets and outputs when using TL, as opposed to using human generated metrics.

References

1. Arutyunov, V., Averkin, A.: Genetic algorithms for music variation on genom platform. Procedia Computer Science **120**, 317–324 (2017). https://doi.org/10.1016/j.procs.2017.11.245
2. Avdeeff, M.: Artificial Intelligence & Popular Music: SKYGGE, Flow Machines, and the Audio Uncanny Valley. Arts **8**(4), 130 (2019). https://doi.org/10.3390/arts8040130
3. Barry, T.: Traditional Scottish Tunes in Midi Format, http://www.whitestick.co.uk/midi.html
4. Brunner, G., Wang, Y., Wattenhofer, R., Zhao, S.: Symbolic music genre transfer with CycleGAN. Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI **2018-Novem**, 786–793 (2018). https://doi.org/10.1109/ICTAI.2018.00123
5. Chen, Z., Wu, C.W., Lu, Y.C., Lerch, A., Lu, C.T.: Learning to fuse music genres with generative adversarial dual learning. Proceedings - IEEE International Conference on Data Mining, ICDM **2017-Novem**, 817–822 (2017). https://doi.org/10.1109/ICDM.2017.98
6. Cope, D.H.: Experiments in Music Intelligence (EMI). Proceedings of the 1987 International Computer Music Conference (1987)
7. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment (sep 2017), http://arxiv.org/abs/1709.06298
8. Dong, H.W., Yang, Y.H.: Convolutional generative adversarial networks with binary neurons for polyphonic music generation. In: Proceedings of the 19th In-

ternational Society for Music Information Retrieval Conference, ISMIR 2018. pp. 190–196 (apr 2018). https://doi.org/10.5281/zenodo.1492377

9. Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., Simonyan, K.: Neural audio synthesis of musical notes with WaveNet autoencoders. 34th International Conference on Machine Learning, ICML 2017 **3**, 1771–1780 (2017)
10. Finnerty, A.: BrightRED Study Guide National 5 Music (2017)
11. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems **3**(January), 2672–2680 (2014)
12. Grall, A., Hamidinekoo, A., Malcolm, P., Zwiggelaar, R.: Using a Conditional Generative Adversarial Network (cGAN) for Prostate Segmentation. In: Communications in Computer and Information Science. vol. 1065 CCIS (2020). https://doi.org/10.1007/978-3-030-39343-4_2
13. Harte, C., Sandler, M., Gasser, M.: Detecting harmonic change in musical audio. Proceedings of the ACM International Multimedia Conference and Exhibition pp. 21–26 (2006). https://doi.org/10.1145/1178723.1178727
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings pp. 1–26 (2018)
15. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2019-June**, 4396–4405 (2019). https://doi.org/10.1109/CVPR.2019.00453
16. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 **2017-Janua**, 105–114 (2017). https://doi.org/10.1109/CVPR.2017.19
17. Lee, S.g., Hwang, U., Min, S., Yoon, S.: Polyphonic Music Generation with Sequence Generative Adversarial Networks (2017), http://arxiv.org/abs/1710.11418
18. Loughran, R., O'Neill, M.: Generative Music Evaluation: Why do We Limit to 'Human'? Proceedings of the 1st Conference on Computer Simulation of Musical Creativity (Ml), 1–16 (2016)
19. Raffel, C.: Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. Phd thesis (2016)
20. Rothstein, J.: Midi Comprehensive Introduction. AR Editions, Inc., 7 edn. (1995)
21. Sandred, Ö., Laurson, M., Kuuskankare, M.: Revisiting the Illiac Suite—a rule-based approach to stochastic processes. Sonic Ideas/Ideas Sonicas pp. 1–8 (2009)
22. Steedman, M.J.: The perception of musical rhythm and metre. Perception **6**(5), 555–569 (1977). https://doi.org/10.1068/p060555
23. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. Advances in Neural Information Processing Systems (Nips), 613–621 (2016). https://doi.org/10.13016/m26gih-tnyz
24. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: Realistic Image Synthesis with Stacked Generative Adversarial Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1947–1962 (2019). https://doi.org/10.1109/TPAMI.2018.2856256
25. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. Proceedings of the IEEE International Conference on Computer Vision **2017-Octob**, 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244