LEARNING TO TRAVERSE LATENT SPACES FOR MUSICAL SCORE INPAINTING

Ashis Pati[1] **Alexander Lerch**[1] **Gaëtan Hadjeres**[2]
[1] Center for Music Technology, Georgia Institute of Technology, Atlanta, USA
[2] Sony CSL, Paris, France
ashis.pati@gatech.edu, alexander.lerch@gatech.edu, gaetan.hadjeres@sony.com

ABSTRACT

Music Inpainting is the task of filling in missing or lost information in a piece of music. We investigate this task from an interactive music creation perspective. To this end, a novel deep learning-based approach for musical score inpainting is proposed. The designed model takes both past and future musical context into account and is capable of suggesting ways to connect them in a musically meaningful manner. To achieve this, we leverage the representational power of the latent space of a Variational Auto-Encoder and train a Recurrent Neural Network which learns to traverse this latent space conditioned on the past and future musical contexts. Consequently, the designed model is capable of generating several measures of music to connect two musical excerpts. The capabilities and performance of the model are showcased by comparison with competitive baselines using several objective and subjective evaluation methods. The results show that the model generates meaningful inpaintings and can be used in interactive music creation applications. Overall, the method demonstrates the merit of learning complex trajectories in the latent spaces of deep generative models.

1. INTRODUCTION

Over the last decade, machine learning techniques have emerged as the tool of choice for the design of symbolic music generation models [1] with deep learning being the most widely used [2]. Deep generative models have been successfully applied to several different music generation tasks, e.g., monophonic music generation [3–5], polyphonic music generation [6, 7] and creating musical renditions with expressive timing and dynamics [8, 9]. However, most of these models assume sequential generation of music, i.e, the generated music depends only on the music that has preceded it. In other words, the models rely only on the past musical context. This approach does not align with typical human compositional practices which are often iterative and non-sequential in nature. In addition, the sequential generation paradigm places severe limitations on the degree of interactivity allowed by these models [10, 11]. Once generated, there is no way to tweak specific parts of the generation so as to conform to users' aesthetic sensibilities or compositional requirements.

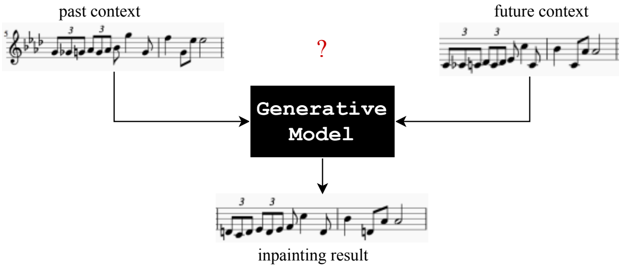

Figure 1: *Musical Score Inpainting* task schematic. A generative model needs to take past and future musical contexts into account to generate a sequence that can connect them in a musically meaningful manner.

In this paper, we seek to address these problems by incorporating future musical context into the generation process. Specifically, the task is to train models to fill in missing information in musical scores, duly taking into account the complete musical context — both past and future. In essence, this is similar to *inpainting* where the objective is to reconstruct missing or degraded parts of any kind of media [12]. For music, inpainting has been traditionally used for restoration purposes [13] or to remove unwanted artifacts such as clipping [14, 15] and packet loss [16]. However, we investigate models for *Musical Score Inpainting* (see Figure 1 and Section 3.1) as tools for music creation which can aid people in (i) getting new musical ideas based on specific styles, (ii) joining different musical sections together, and (iii) modifying or extending solos. In addition, such models can allow interactive music generation by enabling users to change the musical context and get new suggestions based on the updated context.

Our main technical contribution is a novel approach for musical score inpainting which relies on *latent* representation-based deep generative models. These models are trained to compress information from high-dimensional spaces, e.g., the space of all 1-bar melodies, to low-dimensional *latent* spaces. While these latent spaces have been shown to be able to encode hidden attributes of musical data (see Section 2.3), the primary form of interaction with latent spaces has been using simple operations such as attribute vectors [17, 18] or linear interpolations [19, 20]. Using the proposed method (see Section 3), we demonstrate that Recurrent Neural Networks (RNNs) can be trained using latent embeddings to learn complex trajectories in the latent space. This, in turn, is used to predict how to fill in missing measures in a piece of symbolic music. Our secondary contributions are: (i) a stochastic training scheme which helps model training and generalization (see Section 3.4), and (ii) a novel data encoding scheme using uneven tick durations that allows encoding triplets without substantial increase in sequence length (see Section 3.5). The effectiveness of the proposed method is demonstrated using several objective and subjective evaluation methods in Section 4.

2. RELATED WORK

2.1 Audio & Music Inpainting

The first applications of audio inpainting methods were restoration-oriented [13, 14, 16, 21, 22] using different methods such as matrix factorization [14], non-local similarity measures [22] and audio similarity graphs [16]. While these techniques have been useful for audio-based tasks, they are not easily extendable to symbolic music.

For inpainting in the symbolic domain, the early attempts were based on Markov Chain Monte Carlo (MCMC) methods which allowed users to specify certain constraints, e.g., which notes to generate and which to retain [23, 24]. Another approach, proposed by Lattner et al., used iterative gradient descent to force the output of a deep generative model to conform to a specified structural plan [25]. However, methods based on MCMC (which rely on repeated sampling), and those using iterative gradient descent are slow during inference time and hence unsuitable for interactive applications. More recently, Hadjeres et al. proposed the AnticipationRNN framework [10] which used a pair of stacked RNNs to enforce user-defined constraints during inference. This allowed selective regeneration of specific parts of the music (generated or otherwise) using only two forward passes through the RNN-pair and enabled real-time generations.

2.2 Variational Auto-Encoders

The Variational Auto-Encoder (VAE) [26] is a type of generative model which uses an auto-encoding [27] framework; during training, the model is forced to reconstruct its input. The architecture comprises an encoder and a decoder. The encoder learns to map real data-points $\mathbf{x}$ from a high-dimensional data-space X to points in a low-dimensional space Z which is referred to as the *latent* space. The decoder learns to map the latent vectors back to the data-space. VAEs treat the latent vector as a random variable and model the generative process as a sequence of sampling operations: $\mathbf{z} \sim p(\mathbf{z})$, and $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$, where $p(\mathbf{z})$ is a prior distribution over the latent space, and $p(\mathbf{x}|\mathbf{z})$ is the conditional pdf. Variational inference [28] is used to approximate the posterior by minimizing the KL-divergence [29] between the approximate posterior $q(\mathbf{z}|\mathbf{x})$ and the true posterior $p(\mathbf{z}|\mathbf{x})$ by maximizing the evidence lower bound (ELBO) [26]. The training ensures that the reconstruction accuracy is maximized and realistic samples are generated when latent vectors are sampled using the prior $p(\mathbf{z})$.

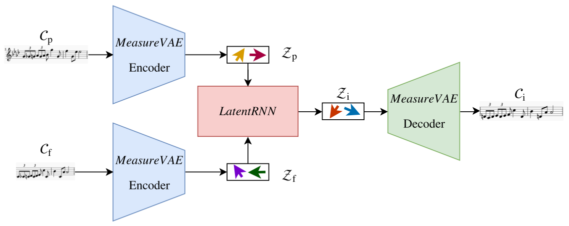

Figure 2: Schematic of the proposed approach. The pre-trained MeasureVAE encoder is used to convert the past and future context sequences ($\mathcal{C}_\mathrm{p}$ and $\mathcal{C}_\mathrm{f}$) into their respective latent vector sequences ($\mathcal{Z}_\mathrm{p}$ and $\mathcal{Z}_\mathrm{f}$). The LatentRNN learns to traverse the latent space of MeasureVAE to output a latent vector sequence $\mathcal{Z}_\mathrm{i}$ which is passed through the pre-trained decoder to output the inpainted musical sequence $\mathcal{C}_\mathrm{i}$.

2.3 Leveraging Latent Spaces for Music Generation

Latent representation-based models such as VAEs have been found to be quite useful for several music generation tasks. Bretan et al. used the latent representation of an auto-encoder-based model to generate musical phrases [30]. Lattner et al. forced the latent space of a gated auto-encoder to learn pitch interval-based representations which improved the performance of predictive models of music [31, 32]. Latent spaces of music generation models have also been used to explicitly encode and control musical attributes [3, 20, 33, 34], inter-track dependencies [35] and musical genre [36]. These studies show that trained latent spaces are able to encode hidden attributes of musical data which can be leveraged for different music generation tasks. However, latent space traversals have been relying on simpler methods such as attribute vectors [17, 18] or linear interpolations [19, 20].

3. METHOD

3.1 Problem Statement

We define the score inpainting problem as follows: given a past musical context $\mathcal{C}_\mathrm{p}$ and a future musical context $\mathcal{C}_\mathrm{f}$, the modeling task is to generate an inpainted sequence $\mathcal{C}_\mathrm{i}$ which can connect $\mathcal{C}_\mathrm{p}$ and $\mathcal{C}_\mathrm{f}$ in a musically meaningful manner. In other words, the model should be trained to maximize the likelihood $p(\mathcal{C}_\mathrm{i} \mid \mathcal{C}_\mathrm{p}, \mathcal{C}_\mathrm{f})$. Without much loss of generality, we assume that $\mathcal{C}_\mathrm{p}$, $\mathcal{C}_\mathrm{f}$, and $\mathcal{C}_\mathrm{i}$ comprise of n_p, n_f, and n_i measures of music, respectively.

3.2 Approach

The key motivation behind the proposed method is that the latent embeddings of deep generative models of music encode hidden attributes of music which can be leveraged to perform inpainting. Firstly, we train a VAE-model, referred to as *MeasureVAE*, to reconstruct single measures of music,

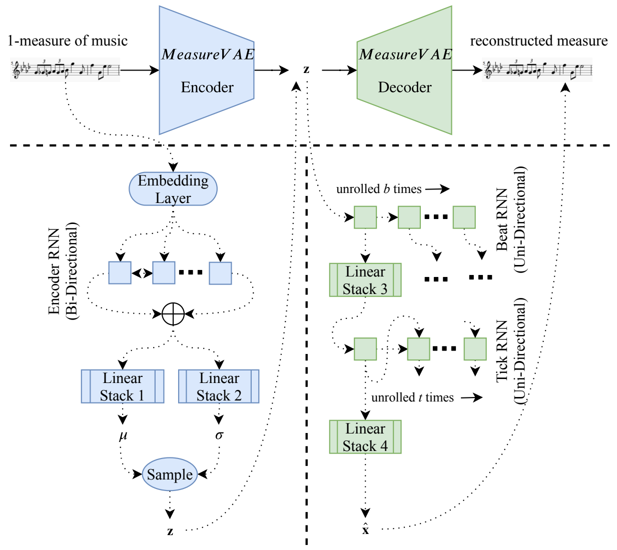

Figure 3: MeasureVAE schematic. Individual components of the encoder and decoder are shown below the main blocks (dotted arrows indicate data flow within the individual components). $\mathbf{z}$ denotes the latent vector and $\hat{\mathbf{x}}$ denotes the reconstructed measure.

i.e., the latent vectors of this model $\mathbf{z} \in Z$ map to individual measures of music. Once trained, the encoder of this model can be used to process sequences $\mathcal{C}_\mathrm{p}$ and $\mathcal{C}_\mathrm{f}$ and output corresponding latent vector sequences $\mathcal{Z}_\mathrm{p}$ and $\mathcal{Z}_\mathrm{f}$. Secondly, we train an RNN-based model, referred to as *LatentRNN*, to take as input the past and future latent vector sequences ($\mathcal{Z}_p$ and $\mathcal{Z}_\mathrm{f}$) and output a third latent vector sequence $\mathcal{Z}_\mathrm{i}$ which can be passed through the decoder of MeasureVAE to obtain $\mathcal{C}_\mathrm{i}$.

Effectively, the LatentRNN model learns to traverse the latent space of the MeasureVAE model so as to connect the provided contexts in a musically meaningful manner. The inference is fast since it only requires forward passes through the two models. This overall approach is shown in Figure 2. We call this joint architecture *InpaintNet*. While we restrict ourselves to $4/4$ monophonic melodic sequences in this paper, the approach can be extended to other time signatures and polyphonic sequences as well. The individual model architectures are discussed next.

3.3 Model Architectures

3.3.1 MeasureVAE

The MeasureVAE architecture (see Figure 3) is loosely based on the hierarchical recurrent MusicVAE architecture [3] which proved successful in modeling individual measures of music.

The encoder consists of a learnable embedding layer (operating on tick-level) followed by a bi-directional RNN [37]. The concatenated hidden state from both directions of the RNN is then passed through two identical parallel linear stacks to obtain the mean μ and variance σ which are used to sample the latent vector $\mathbf{z}$ via $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$.

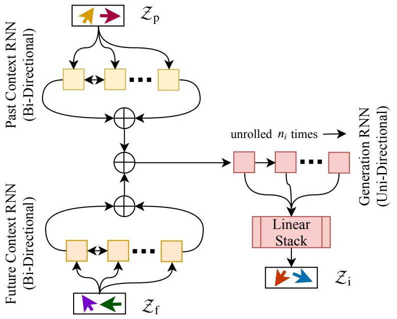

Figure 4: LatentRNN schematic. The Past-Context and Future-Context-RNNs encode $\mathcal{Z}_\mathrm{p}$ and $\mathcal{Z}_\mathrm{f}$, respectively. The Generation-RNN initialized using a concatenation of context-RNNs embeddings is unrolled n_i times to get $\mathcal{Z}_\mathrm{i}$.

The decoder follows a hierarchical structure where the sampled latent vector $\mathbf{z}$ is used to initialize the hidden state of a beat-RNN which is unrolled b times (where b is the number of beats in a measure). The output at each step of the beat-RNN is passed through a linear stack before being used to initialize the hidden state of a tick-RNN which is unrolled t times (where t is the number of events/ticks in a beat). The outputs of the tick-RNN are individually passed through a second linear stack which maps them back to the data-space. The hierarchical architecture mitigates the auto-regressive nature of the RNN and forces the decoder to use the latent vector more efficiently (as advocated in [3]).

3.3.2 LatentRNN

The LatentRNN model (see Figure 4) consists of 3 subcomponents. There are 2 identical bi-directional RNNs, referred to as Past-Context-RNN and Future-Context-RNN, which process the latent vector sequences for the past and future contexts ($\mathcal{Z}_\mathrm{p}$ and $\mathcal{Z}_\mathrm{f}$), respectively. These are unrolled for n_p and n_f times in order to encode the context sequences, respectively. The final hidden states of the two context-RNNs are concatenated and then used to initialize the hidden state of a third RNN, referred to as the Generation-RNN, which is unrolled n_i times. The outputs of the Generation-RNN are passed through a linear stack to obtain n_i latent vectors corresponding to the inpainted measures.

The hyper-parameters for the model configurations are chosen based on initial experiments and are provided in Table 1. For the RNN layers in both models, Gated Recurrent Units (GRU) [38] are used.

3.4 Stochastic Training Scheme

We propose a novel stochastic training scheme for training the model. For each training batch, the number of measures to be inpainted n_i and the number of measures in the past context n_p are randomly sampled from a uniform distribution. Thus, the number of measures in the future context becomes $n_\mathrm{f} = N - n_\mathrm{i} - n_\mathrm{p}$, where N is the total number of measures in each sequence of the training batch. Using these, the input sequences are split into past, future and target sequences and the model is trained to predict

Measure VAE	
Embedding Layer	i=dict size, o=10
EncoderRNN	n=2, i=10, h=512, d=0.5
Linear Stack 1 Linear Stack 2	i=1024, o=256, n=2, non-linearity=SELU
BeatRNN	n=2, i=1, h=512, d=0.5
TickRNN	n=2, i=522, h=512, d=0.5
Linear Stack 3	i=512, o=1024, n=1, non-linearity=ReLU
Linear Stack 4	i=512, o=dict size, n=1, non-linearity=ReLU
Latent RNN	
Past-Context-RNN Future-Context-RNN	n=2, i=256, h=512, d=0.5
Generation RNN	n=2, i=1, h=1024, d=0.5
Linear Stack	i=2048, o=256, n=1, non-linearity=None

Table 1: Table showing configurations of both models. n: Number of Layers, i: Input Size, o: Output Size, h: Hidden Size, d: Dropout Probability, SELU: Scaled Exponential Linear Unit [39], ReLU: Rectifier Linear Unit

the target sequence given the past and future context sequences. This stochastic training scheme ensures that the model learns to deal with variable length contexts and can perform inpaintings at arbitrary locations.

3.5 Data Encoding Scheme

We use a variant of the encoding scheme proposed by Hadjeres et al. [24] for our data representation. The original encoding scheme quantizes time uniformly using the sixteenth note as the smallest sub-division. For each sub-division or tick, the note which starts on that tick is represented by a token corresponding to the note name. If no note starts on a tick, a special continuation symbol '__' is used to denote that the previous note is held. Rest is considered as a note and has a special token. The main advantages of this encoding scheme are (i) it uses only a single sequence of tokens, and (ii) uses real note names (e.g., separate tokens for A# and Bb) which allows generation of readable sheet music.

However, a limitation of using the sixteenth note as the smallest sub-division is that it cannot encode triplets. The naive approach of evenly subdividing the sixteenth note divisions to encode triplets increases the sequence length a factor of 3 which can make the sequence modeling task harder. To mitigate this limitation, we propose a novel uneven subdivision scheme. Each beat is divided into 6 uneven ticks (shown in Figure 5). This allows encoding triplets while only increasing the sequence length by a factor of 1.5. Consequently, each $4/4$ time signature measure is a sequence of 24 tokens.

4. EXPERIMENTS

The proposed method is compared with two baseline methods (see Section 4.1) using a dataset of monophonic folk melodies in the Scottish and Irish style taken from the Session website [5]. For the purposes of this work, only melodies with $4/4$ time signature in which the shortest note is greater than or equal to the sixteenth note are considered resulting in approx. 21000 melodies. Implementation details and source code are available online. [1]

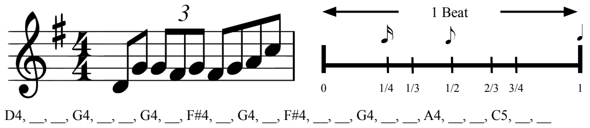
3
1 Beat
0
1/4
1/3
1/2
2/3
3/4
1
D4, __, __, G4, __, __, G4, __, F#4, __, G4, __, F#4, __, __, G4, __, __, A4, __, __, C5, __, __

Figure 5: Figure showing the data representation. The token string on the bottom demonstrates the encoding scheme for the measure displayed on the top-left. Top-right shows the proposed uneven tick-duration scheme for each beat.

4.1 Baseline

The performance of the proposed method is compared with the AnticipationRNN model proposed by Hadjeres et al. [10]. This model, referred to as *Base-ARNN*, uses a stack of 2 LSTM-based [40] RNN layers. Each of the 2 RNNs comprises of 2 layers with a hidden size of 256. In addition to the note-sequence tokens, this model also uses additional metadata information, i.e., tokens to indicate beat and downbeat locations as part of the user-defined constraints. For more details, the readers are directed to [10].

The original model operates on tick-level sequences and inpainting locations are specified in terms of individual tick locations. Hence, the inpainting locations may or may not be contiguous. In order to make a fair comparison, a second variant of the AnticipationRNN model is considered, referred to as *Reg-ARNN*, where the stochastic training scheme from Section 3.4 is used instead.

4.2 Training Configuration

The MeasureVAE model was pre-trained using single measures following the standard VAE optimization equation [26] with the β-weighting scheme [41,42]. In order to prioritize high reconstruction accuracy, a low value of $\beta = 1\mathrm{e}{-3}$ was used. Pre-training was done for 30 epochs resulting in a reconstruction accuracy of approx. 99%. While this seems to be better than results in [3], we attribute this to the shorter duration of generation (single measures) and the differences in datasets and data encoding. MeasureVAE parameters were frozen after pre-training and no gradient-based updates were performed on these parameters during the InpaintNet model training.

The Adam algorithm [43] was used for model training, with a learning rate of $1\mathrm{e}{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1\mathrm{e}{-8}$. To ensure consistency, all models were trained for 100 epochs (with early-stopping) with the same batch-size using a sub-sequence length of 16 measures (384 ticks). For the InpaintNet and Reg-ARNN models, the number of measures to be inpainted and the number of past measures were randomly selected: $n_\mathrm{i} \in [2, 6]$, $n_\mathrm{p} \in [1, 16 - 1 - n_\mathrm{i}]$. This ensured that past and future contexts each contain at

[1] https://github.com/ashispati/InpaintNet

Model Variant	Test NLL
Base-ARNN	0.662
Reg-ARNN	0.402
InpaintNet (Our Method)	**0.300**
PastInpaintNet	0.643
FutureInpaintNet	0.481

Table 2: Table showing the average token-wise NLL (nats/token) on the held-out test set (lower is better). InpaintNet outperforms both baselines. The last two rows show the results for the ablation models described in Section 4.4.

least 1 measure. For the baseline models, teacher-forcing was used with a probability of 0.5.

4.3 Predictions on Test Data

Two experiments were conducted to evaluate the predictive power of the models.

The first experiment considered the average token-wise negative log-likelihood (NLL) on a held-out test set. The results (see first 3 rows of Table 2) indicate that our proposed model outperforms both baselines, showing an improvement of approx. 25% in the NLL over the Reg-ARNN model and approx. 55% over the Base-ARNN model.

The next experiment compared the models by varying the number of measures to be inpainted. Figure 6 shows the average token-wise NLL when n_{i} was increased from 2 to 8. Again, our proposed model outperforms both baselines. It should be noted that since the sub-sequence length is constant at 16 measures, increasing n_{i} means that the available context is reduced. Thus, there is an expected drop in the performance with increasing n_{i} as the models are forced to make longer predictions with less contextual information. However, the InpaintNet model performs better even when forced to predict beyond the training limit of 6 measures.

4.4 Ablations Studies

In order to further ascertain the efficiency of the proposed approach, ablation studies were conducted to evaluate the benefit of adding past and future context information. Specifically, we trained two variants of the InpaintNet model which relied on only one type of contextual information. The first model, referred to as PastInpaintNet only considered the past context $\mathcal{C}_{\mathrm{p}}$ as input whereas the second model, referred to as FutureInpaintNet considered only the future context $\mathcal{C}_{\mathrm{f}}$. The last two rows of Table 2 summarize the performance of these ablation models. It is clear that both past and future contexts are important for the modeling process. In addition, we also tried training a variant of the InpaintNet model with an untrained (randomly initialized) MeasureVAE model. This model failed to train properly achieving an NLL of approx. 1.33. This indicates that a structured latent space where latent vectors are trained to encode hidden data attributes is important for training the LatentRNN model.

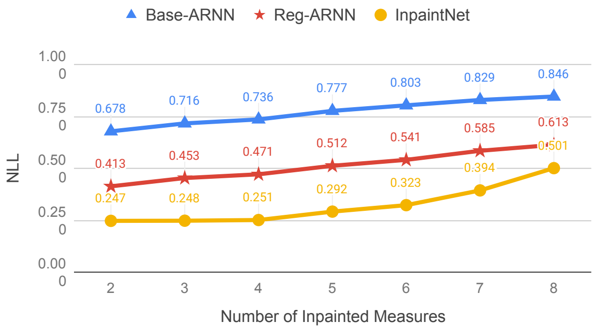

Figure 6: Figure showing token-wise NLL (nats/token) for different number of inpainted measures on the held-out test set (lower is better). InpaintNet outperforms both baselines. Models were trained to predict only 2 to 6 measures.

4.5 Qualitative Analysis

Considering that we are primarily interested in the aesthetic quality of the inpaintings, we encourage the readers to browse through the inpainting examples provided in the supplementary material. [2] We consider some of those examples in the analysis below.

Figure 7 shows sample inpaintings by the models for one of the melodies in the test set. While the Base-ARNN model collapses to produce long half notes which do not effectively reflect the surrounding context, the other two models do better. Both the Reg-ARNN and InpaintNet model generate rhythmically consistent inpaintings. The InpaintNet, in particular, mimics the rhythmic properties of the context better. For instance, measures 7 and 10 of the inpainted measures match the rhythm of measures 6, 14, and 15. Also, measure 8 matches measure 16. However, the use of G (subdominant scale degree in D-major) in the half-note to end measure 8 is unusual. We observed that in other examples also, the InpaintNet model occasionally produces pitches which are anomalous — either out-of-key or not fitting in the context. The Reg-ARNN model, on the other hand, tends to stay in key. Additional examples are provided in the supplementary material.

One advantage of working with the latent space is that the sampling operation, inherent in the VAE inference process, ensures that for the same context we can get different inpainting results. Figure 8 shows three such generations for the context of Figure 7. It is interesting to note that the base rhythm is retained across all three inpaintings. This feature is particularly interesting from an interactive music generation perspective, as this model can be used to quickly provide users with multiple ideas and will be investigated further in future work.

4.6 Subjective Listening Study

To evaluate the perceived quality of the inpainted measures, a listening test was conducted to compare our proposed model against the two baselines. A set of 30 melodies from the held-out test set were randomly selected and their first

[2] https://ashispati.github.io/inpaintnet/

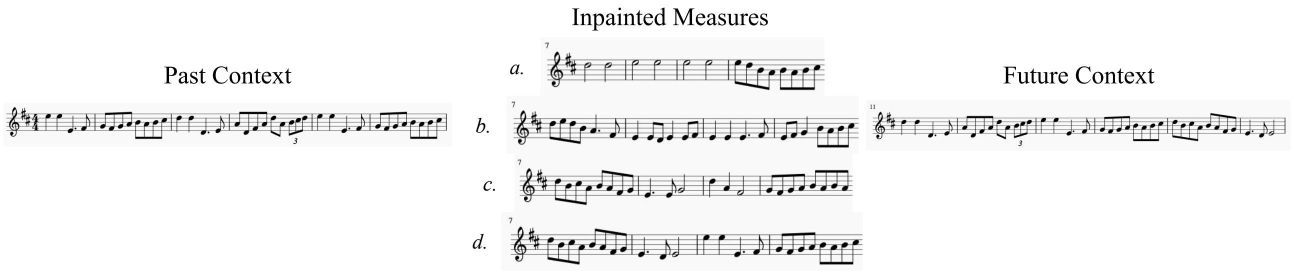

Figure 7: Figure showing the inpaintings generated by different models for the same context. From top to bottom — *a.*: Base-ARNN, *b.*: Reg-ARNN, *c.*: InpaintNet, *d.*: Original Melody.

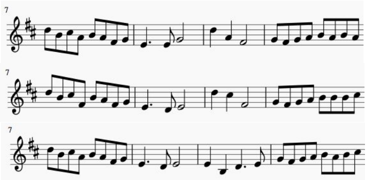

Figure 8: Figure showing different inpaintings (using the InpaintNet model) for the same context as Figure 7.

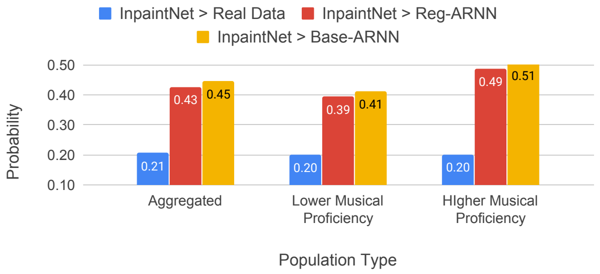

Figure 9: Figure based on the subjective listening study showing the probability that the InpaintNet model is rated higher. Analysis is based on the Bradley-Terry model [44, 45]. The proposed model loses against the real data but performs at par with the baseline models.

16 measures were extracted. The models were then used to inpaint 4 measures (measure number 7 to 10) in these melodic excerpts. Participants were presented with pairs of melodic excerpts and asked to select the one in which they thought the inpainted measures fit better within the surrounding context. In some of the pairs, one melodic excerpt was the real data (without any inpainting). Each participant was presented with 10 such pairs. A total of 72 individuals participated in the study (720 comparisons). The location of the inpainted measures was kept consistent across all examples so as to prevent confusion among participants and allow them to focus better on the inpainted measures.

The Bradley-Terry model [44,45] for paired comparisons was used to get an estimate of how the proposed model performs against the baselines and the real data (see Figure 9). While the proposed model expectedly has a very low probability of winning against the real data (wins approx. 1 out of 5 times), it performs only at par with the baseline models (with probability approx. 0.5). Significance tests using the Wilcoxon signed rank test were further conducted which validated that differences between the proposed model and the baselines were not statistically significant (p-value > 0.01). This was unexpected since the proposed model showed significant improvement over the baselines in the NLL metric. Further dividing the study population into two groups differing in musical proficiency (based on the Ollen index [46]) showed that, comparatively, the group with greater musical proficiency favored the generations from the InpaintNet model more than the group with less musical proficiency.

Additional analysis revealed that cases where the InpaintNet model performed the worst (maximum losses against the baselines), had anomalies in the predicted pitch similar to those discussed in Section 4.5. Specifically, they either had a single out-of-key note (e.g., F note in G-Major scale) or used a pitch or interval not used in the provided contexts. We conjecture that it is these anomalous pitch predictions which lead to poor perceptual ratings in spite of the model performing better in terms of modeling rhythmic features. This will be analyzed further in future studies.

5. CONCLUSION

This paper investigates the problem of musical score inpainting and proposes a novel approach to generate multiple measures of music to connect two musical excerpts by using a conditional RNN which learns to traverse the latent space of a VAE. We also improve upon the data encoding and introduce a stochastic training process which facilitate model training and improve generalization. The proposed model shows good performance across different objective and subjective evaluation experiments. The architecture also enables multiple generations with the same contexts, thereby, making it suitable for interactive applications [47]. We think the idea of learning to traverse latent spaces could be useful for other music generation tasks also. For instance, the architecture of the LatentRNN model can be changed to add contextual information from other voices/instruments to perform multi-instrument music generation. Future work will include a more thorough investigation of the anomalies in pitch prediction. A possible way to address that would be to add the context embedding as input at each step of unrolling the LatentRNN or use additional regularizers. Another promising avenue for future work is substituting RNNs with attention-based models [48] which have had success in sequential music generation tasks [9].

6. REFERENCES

[1] Rebecca Fiebrink, Baptiste Caramiaux, R Dean, and A McLean. *The machine learning algorithm as creative musical tool*. Oxford University Press, 2016.

[2] Jean-Pierre Briot and François Pachet. Deep learning for music generation: Challenges and directions. *Neural Computing and Applications*, Oct 2018.

[3] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical hatent vector model for learning long-term structure in music. In *Proc. of the 35th International Conference on Machine Learning (ICML)*, pages 4364–4373, Stockholmsmässan, Stockholm Sweden, 2018.

[4] Florian Colombo, Samuel P. Muscinelli, Alexander Seeholzer, Johanni Brea, and Wulfram Gerstner. Algorithmic composition of melodies with deep recurrent neural networks. In *Proc. of the 1st Conference on Computer Simulation of Musical Creativity (CSMC)*, 2016.

[5] Bob L Sturm, Joao Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. Music transcription modelling and composition using deep learning. In *Proc. of the 1st Conference on Computer Simulation of Musical Creativity (CSMC)*, Huddersfield, UK, 2016.

[6] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. In *Proc. of International Society of Music Information Retrieval Conference (ISMIR)*, pages 324–331, Suzhou, China, 2017.

[7] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proc. of 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012.

[8] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, pages 1–13, 2018.

[9] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. In *Proc. of International Conference of Learning Representations (ICLR)*, New Orleans, USA, 2019.

[10] Gaëtan Hadjeres and Frank Nielsen. Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation. *Neural Computing and Applications*, Nov 2018.

[11] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation-A survey. *arXiv preprint arXiv:1709.01620*, 2017.

[12] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[13] Amir Adler, Valentin Emiya, Maria G Jafari, Michael Elad, Rémi Gribonval, and Mark D. Plumbley. Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):922–932, 2012.

[14] Çağdaş Bilen, Alexey Ozerov, and Patrick Pérez. Audio declipping via nonnegative matrix factorization. In *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE, 2015.

[15] Christopher Laguna and Alexander Lerch. An efficient algorithm for clipping detection and declipping audio. In *Proc. of the 141st AES Convention*, Los Angeles, USA, 2016.

[16] Nathanael Perraudin, Nicki Holighaus, Piotr Majdak, and Peter Balazs. Inpainting of long audio segments with similarity graphs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(6):1083–1094, 2018.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119, 2013.

[18] Shan Carter and Michael Nielsen. Using artificial intelligence to augment human intelligence. *Distill*, 2017. https://distill.pub/2017/aia.

[19] Adam Roberts, Jesse Engel, Sageev Oore, and Douglas Eck. Learning latent representations of music to generate interactive musical palettes. In *Proc. of IUI Workshops*, 2018.

[20] Gaëtan Hadjeres, Frank Nielsen, and François Pachet. GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. In *Proc. of IEEE Symp. Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[21] Çağdaş Bilen, Alexey Ozerov, and Patrick Pérez. Joint audio inpainting and source separation. In *Proc. of International Conference on Latent Variable Analysis and Signal Separation*, pages 251–258. Springer, 2015.

[22] Ichrak Toumi and Valentin Emiya. Sparse non-local similarity modeling for audio inpainting. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 576–580. IEEE, 2018.

[23] Jason Sakellariou, Francesca Tria, Vittorio Loreto, and François Pachet. Maximum entropy model for melodic patterns. In *Proc. of the ICML Workshop on Constructive Machine Learning*, Lille, France, 2015.

[24] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. DeepBach: A steerable model for Bach chorales generation. In *Proc. of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1362–1371, Sydney, Australia, 2017.

[25] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints. *Journal of Creative Music Systems*, 2, March 2018.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of International Conference of Learning Representations (ICLR)*, Banff, Canada, 2014.

[27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the 25th International Conference on Machine learning (ICML*, pages 1096–1103, Helsinki, Finland, 2008.

[28] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[29] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[30] Mason Bretan, Gil Weinberg, and Larry Heck. A unit selection methodology for music generation using deep neural networks. In *Proc. of the 8th International Conference on Computational Creativity (ICCC)*, Atlanta, USA, 2016.

[31] Stefan Lattner, Maarten Grachten, and Gerhard Widmer. A predictive model for music based on learned interval representations. In *Proc. of International Society of Music Information Retrieval Conference (ISMIR)*, pages 26–33, Paris, France, 2018.

[32] Andreas Arzt and Stefan Lattner. Audio-to-score alignment using transposition-invariant features. In *Proc. of International Society of Music Information Retrieval Conference (ISMIR)*, pages 592–599, Paris, France, 2018.

[33] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *Proc. of International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[34] Ashis Pati and Alexander Lerch. Latent space regularization for explicit control of musical attributes. In *ICML Machine Learning for Music Discovery Workshop (ML4MD), Extended Abstract*, Long Beach, CA, USA, 2019.

[35] Ian Simon, Adam Roberts, Colin Raffel, Jesse Engel, Curtis Hawthorne, and Douglas Eck. Learning a latent space of multitrack measures. In *Proc. of the 2nd Workshop on Machine Learning for Creativity and Design*, Montréal, Québec, 2018.

[36] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. In *Proc. of International Society of Music Information Retrieval Conference (ISMIR)*, pages 747–754, Paris, France, 2018.

[37] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[38] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proc. of 32nd International Conference on Machine Learning (ICML)*, pages 2342–2350, Lille, France, 2015.

[39] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 971–980, 2017.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[41] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-Vae: Learning basic visual concepts with a constrained variational framework. In *Proc. of International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[42] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proc. of the 20th Conference on Computational Language Processing*, 2015.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[44] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[45] David R Hunter et al. MM algorithms for generalized Bradley-Terry models. *The annals of statistics*, 32(1):384–406, 2004.

[46] Joy E Ollen. *A criterion-related validity test of selected indicators of musical sophistication using expert ratings*. PhD thesis, The Ohio State University, 2006.

[47] Théis Bazin, Ashis Pati, and Gaëtan Hadjeres. A model-agnostic web interface for interactive music composition by inpainting. Neural Information Processing Systems (NeurIPS), 2018. Demonstration Track.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.