Music Style Transformer

Music Generation via Raw Audio Transcription with Applications to Style-Transfer

Jason St. George
Department of Computer Science
Rochester Institute of Technology
Rochester, New York, USA
jps9013@rit.edu

Hans-Peter Bischof
Department of Computer Science
Rochester Institute of Technology
Rochester, New York, USA
hpb@cs.rit.edu

***Abstract*—Creating salient musical generative models in the neural network community has proven a notoriously difficult task. Music requires the coherent modeling of structure at many different timescales, giving rise to complex hierarchical-temporal structure and self-similarity. Representing music as a discrete sequence of tokens (notes), as in Natural Language modeling, we invoke a Bayesian Framework to factorize the problem into three components which can transcribe, compose, and synthesize audio waveforms of lengths up to ∼100sec, based on work done by the Magenta project [1]. Toward a coherent neural model with applications to Style Transfer, we train a suite of models on a series of curated, genre-specific datasets, some of which an order of magnitude larger than the MAESTRO dataset. We evaluate models quantitatively and qualitatively using information theory, a single-blind survey of model comparisons, visualization techniques, and a novel similarity measure.**

***Index Terms*—Automatic Music Transcription; Relative Self-Attention; Transformer; Music Generation; Style Transfer**

I. Introduction

Music generation has gained increasing attention with the onset of deep learning and the third generation of machine learning in conjunction with the increase in compute power over the preceding decades. Various attempts have been made to produce music using machine learning techniques. Notable efforts have typically taken either a sequential approach, primarily utilizing RNNs and LSTMs as in BachBot [2], non-Hierarchical Convolutional models like Counterpoint By Convolution [3], and Hierarchical Encoder-Decoder models such as MusicVAE [4] which samples from an encoded distribution for generation. For highly restrictive environments, many sequence modeling approaches produce impressive results that are difficult to distinguish from human-generated content, however with the cost of generality.

Many of these approaches have several drawbacks. Recurrent models typically fail to impose a hierarchical ordering with long-term dependencies and thus are unable to generate coherent sequences longer than a few seconds. Non-recurrent models do not capture relative or absolute positions along the time dimension and thus require explicit encoding of positional information to utilize sequential ordering of inputs [5]. Convolutional models implicitly capture relative positions within the kernel, however have been shown to still benefit from positional encodings [6]. Encoder-Decoder models often will have 'holes' in the latent space, such that decoding a random sample may result in nothing realistic to the human listener. The majority of methods simplify the input data representation, reducing the complexity of the input and thus the representational power of the model.

To overcome these limitations, a Wave to MIDI transcription framework was proposed by Hawthorne et al. (2018) [7] in which raw audio waveforms are transcribed and converted to MIDI, encoded and passed to a Transformer [8] model with a Relative Attention [5] mechanism for generation, whose output is then passed to a Conditional WaveNet model to decode the intermediate generated representation and output a waveform audio signal. The Music Transformer injects hierarchical structure and introduces self-reference to the generative process, such that longer coherent sequences can be generated in a semi-supervised setting [9].

Contributions of this paper

The main contribution of this paper lie mainly in the domain of style transfer and the application of evaluation tools from cognitive computing. The objective has three components; the first is to aggregate curated and customized datasets into two main categories: *Genres* and *Composers*.

These data are taken from a private collection totaling over 2,200 hours of polyphonic piano music. Several datasets were created in each category so as to compare results of model-generated examples via quantitative and qualitative measurements, including a single-blind sample study of both lay listeners and musicians. Thus, the second goal is the application of a set of evaluation metrics to further understanding of relationships between compositional styles and human perceptible differences in the context of hierarchical-temporal structure.

Style Transfer dataset examples:

1) *Single Composer:* Bach, Beethoven, Brahms, Chopin, Debussy, Keith Jarrett, Liszt, Brad Mehldau, Scriabin, Art Tatum
2) *Single Genre:* Classical only, Jazz only
3) *Mixed Genre:* Classical and Jazz combined

We intend to make these datasets publicly available in TFRecord format for reproducibility and to widen the scope of information available to researchers in the near future. Until then, please contact the primary author for inquiries regarding obtaining data for research or personal use.

II. Related Work

A motivational insight by Engel et al. (2017) [10] is to explicitly factorize the generation of music into notes and other musical qualities, allowing for modularization of the task.

$$P(audio) = P(audio|note)P(note). \tag{1}$$

Model Overview

This Bayesian factorization (1) of the probability distribution of audio waveforms can be further specified to create three modular task models, noted by Huang et al. (2018),

$$P(audio) = \mathbb{E}_{notes}[P(audio|notes)]. \tag{2}$$

The task of each model is described in detail in [7]. They are as follows:

1) *Encoder,* $P(notes|audio)$: Transcription, takes raw audio in WAV format as input and outputs a MIDI transcription of the raw audio signal, modeled for piano (Hawthorne et al, 2017).
2) *Prior,* $P(notes)$ Generative, uses a relative self-attention mechanism to take as input a transcribed MIDI dataset converted to a Performance Encoding [11] and produces generated MIDI as output when the learned distribution is sampled from [9].
3) *Decoder,* $P(audio|notes)$ Synthesis model, converts generated MIDI samples from the transformer to a reconstructed raw audio format (WAV) [12].

Piano-e-Competition and MAESTRO

A recent development is the release of the MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) dataset [7] which contains piano performances captured from several years of the International Piano-e-Competition, with approximately 3ms accuracy fine alignment [13] between note labels and audio waveforms. The MAESTRO dataset is over an order or magnitude larger than previously available datasets. These properties make the dataset desirable, most notably homogeneity; all performances are solo piano, mostly one genre (classical), and all played by expert humans [14].

Automatic Music Transcription

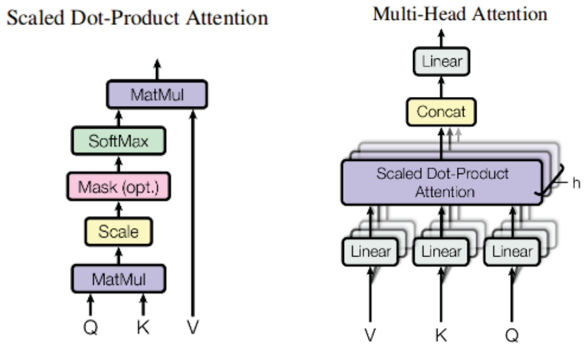
Automatic Music Transcription (AMT) has the goal of creating a symbolic representation of music from raw audio. Commonly, MIDI is a desirable representation as it is simple, compact, easily interpretable, and standardized. The goal of many AMT tasks is to generate transcriptions of audio which contains all perceptually relevant performance information without prior knowledge, such as recording environment or instrument characterization.

Advancing state of the art in polyphonic piano music transcription, Roberts et al. (2017) [15] use a deep convolutional and recurrent neural network which is trained to jointly predict onsets and frames. Their model predicts pitch onset events which subsequently condition framewise pitch predictions, building on previous models designed for tractably modeling the distribution of images as a product of conditional distributions such as Pixel RNN [16] using a two-dimensional LSTM architecture proposed by Theis and Bethge (2015) [17], and PixelCNN++ [18]. Further building on PixelRNN, the authors add a discretized logistic mixture likelihood for modeling the distribution of sub-pixel values with improved memory complexity while simultaneously avoiding gradient saturation via the 256-way categorical softmax function in PixelRNN. The Onsets model also predicts velocities of normalized audio which translates to the speed of attack of a note. Enabled by the precise velocity timings recorded in the Piano-e Competition dataset, the addition of velocity prediction results in more accurate transcriptions of piano music.

We use the Onsets model as the first step in the Bayesian framework as an encoder to convert raw audio into an intermediate representation (MIDI) that is then consumed by the Music Transformer decoder model for prediction.

Attention Model

Modeling long-term dependencies has been a key challenge in many sequence transduction tasks [19]. The most important factor to consider is the length of paths forward and backward signals have to traverse in a given network. The shorter the signal paths between dependencies that must traverse through the network, the easier it is to learn.



((a)) Scaled dot product attention ((b)) Multi-head attention

Fig. 1. Attention mechanism

The Transformer is an autoregressive encoder-decoder model that utilizes self-attention mechanisms and sinusoidal position information to encode distance, achieving state of the art in machine translation tasks. The model consists of stacked encoder-decoder layers, each with separate sub-layers. A self-attention layer in the encoder sub-layer is followed by a positional feed-forward layer, whereas in the decoder sub-layer there is an additional masking operation so as to not allow previous output from incorporating information about future output positions during training. Each of the layers contain batch normalization [20]. For more implementation details, see [8] [9]. Attention can be described as mapping a set of query and key-value pair vectors, whose output is computed as a weighted sum of the values. The weighting of each output is calculated by a compatibility function on the query with its corresponding key. *Scaled Dot-Product Attention* (Fig 1a) consists of queries and keys of dimension d_k, and values of dimension d_v. The dot products of the queries are computed with all keys, each scaled by $\sqrt{d_k}$, and then applying the softmax function to obtain the weights on the values. This weighting function is separated into h attention heads, allowing each head to focus on a particular subset of the input data. In addition to learning different tasks, each head may learn meaningful semantic information [8]. The attention function computes queries simultaneously, packaged into query matrix Q, key matrix K, and value matrix V. The attention output for a given head Z_h is given by

$$Z_h = \text{softmax}\left(\frac{(Q^h K^h)^\top}{\sqrt{D_h}}\right)V^h. \tag{3}$$

where h refers to a single attention head, such that each head works on a distinct subset of the entire input $x_i \in \mathbb{R}^{d_n}$ of n elements.

Each head computes a new sequence $z_i \in \mathbb{R}^{d_n}$,

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V) \tag{4}$$

$$\alpha_{ij} = \frac{\text{exp } e_{ij}}{\sum_{k=1}^{n} \text{exp } e_{ik}} \tag{5}$$

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^\top}{\sqrt{d_z}} \tag{6}$$

where $W^Q, W^K, W^K \in \mathbb{R}^{d_x \times d_z}$ are parameter matrices.

Without recurrent or convolutional connections, the model cannot explicitly utilize the order of the sequence or relative positional information for training and must be inserted. Positional encodings are mapped using sine and cosine functions of different frequencies to model the relative distance between adjacent positions of the input sequence:

$$PE_{pos,2i} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{7}$$

$$PE_{pos,2i+1} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{8}$$

where *pos* is position and *i* is the dimension, where each dimension of the positional encoding corresponds to a sinusoid. This is reasonable, since for any fixed offest k, PE_{pos+k} can be represented as a linear function of PE_{pos}

Self-attention layers connect all positions with a constant number of sequentially executed operations, whereas recurrent layers require linear, $\mathcal{O}(n)$. A convolutional layer with kernel size $k < n$ does not connect all pairs of input and output positions, but would require a stack of $\mathcal{O}(\frac{n}{k})$ layers of contiguous kernels, or $\mathcal{O}(log_k(n))$ dilated convolutional layers [8]. Attention also greatly improves the interperatibility of models, since each attention head learns different tasks and exhibit syntactic and semantic structural elements upon inspection of learned parameter distributions.

Relative Positional Self-Attention

Shaw et al. (2017) [5] extend the Transformer model to efficiently incorporate representations of the relative positions between sequential elements of the input to consider arbitrary relationships between any two elements. An edge in the graph connecting input elements x_i and x_j is represented by vectors $a_{ij}^K, a_{ij}^V \in \mathbb{R}^{d_a}$. By modeling the input as a labeled, directed and fully-connected graph whose representations are shared across attention heads, edge information contained in a_{ij}^K, a_{ij}^V can be propagated to sub-layer outputs a weighted sum of linearly transformed input tokens z_i,

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V + a_{ij}^V) \tag{9}$$

The compatibility function from [8] is extended to consider edges, modifying (6) to obtain

$$e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K)^\top}{\sqrt{d_z}} \tag{10}$$

For a more efficient implementation, Shaw et al. split the numerator from (10) into two terms and perform tensor reshaping,

$$e_{ij} = \frac{x_i W^Q (x_j W^K)^\top + x_i W^Q (a_{ij}^K)^\top}{\sqrt{d_z}} \tag{11}$$

Dropping the h index for clarity, for each head we have

$$Relative\ Attention = softmax\left(\frac{QK^\top + Q(A^K)^\top}{\sqrt{D_h}}\right)V \tag{12}$$

where matrix A contains all entries of a_{ij}^K from (11), and

$$softmax(z) = \frac{e^z}{\sum_{i=1}^{n} e_i^z} \tag{13}$$

Huang et al. (2018) [9] implement a skewing procedure that improves the intermediate memory requirement from $\mathcal{O}(L^2 D)$ to $\mathcal{O}(LD)$ while maintaining relative position embeddings and doing away with sinusoidal-based position functions.

Audio Synthesis

Breakthroughs in modeling complex distributions of images [18] and text [21] using fully probabilistic and autoregressive generative models have been adapted to the domain of audio signals by modeling the joint probabilities over signals as products of conditional distributions, leading to state-of-the-art generation and synthesis [12].

WaveNets use stacked *dilated causal convolutions*, enabling very large receptive fields with few layers (Fig 2) and *Gated Activation Units* (Fig 3) [18] with residual and skip connections to model the conditional probability distribution of audio given all previous samples as

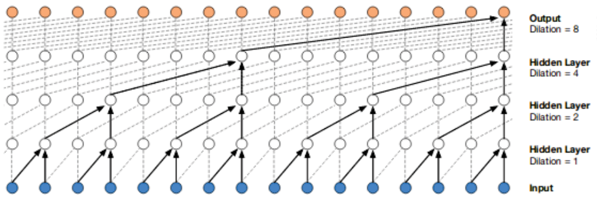


Fig. 2. WaveNet dilated causal convolutions

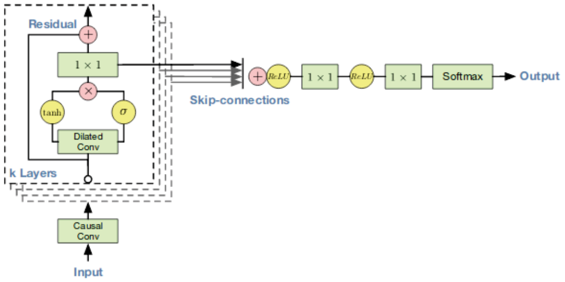


Fig. 3. WaveNet residual connections with gated activation units

$$p(x|h) = \prod_{t=1}^{\top} p(x_t|x_1,...,x_{t-1},\mathbf{h}). \tag{14}$$

Dilated causal convolution stacks allow WaveNet's receptive fields to grow exponentially with depth. WaveNets can produce audio with specified characteristics such as speaker identity by conditioning on different input variables, allowing incredible flexibility in reproducing desired acoustic and spectral parameters achieving state-of-the-art results in TTS (Text-To-Speech).

III. Data Representation

Encoding polyphonic music as a single serialized stream of discrete tokens allows a language-modeling approach to learning from data. In order to get these data into a workable format for modeling symbolic music, MIDI is often used as an intermediary representation for compactness and flexibility.

MIDI

The standard format MIDI (Musical Instrument Digital Interface) is represented as a set of tokens from the following vocabulary of size 413 to account for all possible values:

1) **128 NOTE-ON** events: Beginning of a note (onset) within specified MIDI pitch-range.
2) **128 NOTE-OFF** events: End of a given note.
3) **125 TIME-SHIFT** events: moves the time-step forward in increments of 8ms up to 1 second.
4) **32 VELOCITY** events: Alters the velocity (speed of attack) applied to all subsequent notes until the next velocity event.

MIDI files are often visualized as a piano roll, in which the y-axis represents the space of all possible notes within a given range and the x-axis represents time. To utilize MIDI as input to the generative model, we apply several transformations on the transcribed waveform data such that the end result is a binary encoded one-hot vector representation of the content of the MIDI files [11]. The stages are as follows:

1) Convert MIDI to Note Sequence Protocol
2) Data Augmentation (Stretching and Transposition)
3) Convert augmented data to Performance Encoding

Note Sequence Protocol

Note sequence protocol is an extensible language and platform agnostic method developed by Google Brain for serializing structured data, similar to XML [22]. Refer to Huang et al. (2018) and the Google Protocol Buffer github for specific implementation details [9].

Performance Encoding

We use the *performance encoding* [11] which serializes MIDI sequences that were converted to performance indices in Figure 8 into one-hot vectors $M \in \mathbb{R}^{d_n \times d_m}$ where d_n is the number of events in a given sequence and $d_m = 413$, for one of the 413 potential MIDI events that can occur in a quantized unit of time.

Data Augmentation

As proposed by Oore et al. (2018) [11], we apply a two specialized data augmentation functions to increase the training set size using symmetrical transformations on the data, with small modifications.

1) *Transposition*: Each example is transposed ± 4 semitones, the distance of a major third, yielding 8 new examples.
2) *Stretching*: Each example is time-dilated uniformly by $\pm 2.5\% and \pm 5\%$, yielding 4 new examples.

IV. Training

We use a similar regime for each model as in [9], training each model between 1000000 and 1250000 steps on a single GTX 1070i with default Transformer hyperparameters [8], a learn rate of 0.1, dropout at 0.1, local attention block size of 512, and target sequence lengths of 1024, 2048, and 4096.

Decoding works similarly, as we sample from the model and can control the decode sequence length of output to be generated and then convert to MIDI for inspection.

V. Evaluation

Quantitative

We utilize concepts from information theory and statistical learning theory to quantitatively evaluate our models, namely Information Rate (IR), Negative Log Likelihood (NLL), and Negative Log Perplexity (NLP).

As entropy can also be thought of as a measure of the predictability of an event in a given context, evaluating Information Content over the time-dimension in generated sequences may constitute a sufficient metric for evaluating musical structure [23]. This motivates our use of Information Rate an extension of Shannon's work as well as Perplexity as a means of characterizing the complexity of a stochastic sequence.

Negative Log Perplexity: Perplexity is the exponentiation of the entropy, where we use entropy as a measure of the expected number of bits required to encode the outcome of the random variable.

In order to adapt the perplexity metric to our model, we use the following formulation on N test data points:

$$2^{-\sum_{i=1}^{N}\frac{1}{N}\log_2 q(x_i)}$$

Negative Log Likelihood: By definition, *minimizing* KL-divergence is equivalent to *maximizing* the likelihood of our learned parameters and thus *minimizing the distance* between the two distributions,

$$\hat{\theta}_{MLE} = \arg\max_{\theta}\sum_{i=1}^{n} logf(x_i|\theta) \qquad (15)$$

We report this common metric for evaluating machine learning models and other in Appendix IV.

Information Rate: Toward defining a salient metric for self-similarity, a critical property of music to examine is the balance between repetition and variation. This ratio can be expressed as a measure taken from Information Theory [24] called Information Rate (19). It can be considered the mutual information (18) between the present and the past observations, calculated as the average difference between the marginal entropy (16) and the conditional entropy (17) of sequences X and Y [25].

Entropy is defined as the expectation of the information content of the random variable

$$H(X) = \mathbb{E}[I(X)] = -\sum_{x} p(x)\ \log_2[p(x)] \qquad (16)$$

We approximate *Conditional Entropy* using a first-order Markov chain with asymptotic distribution μ and transition matrix P, and reduce to a stationary stochastic process [26]

$$H(X|Y) = -\sum_{ij}\mu_i P_{ij}\log_2 P_{ij} \qquad (17)$$

Mutual Information is then defined as the difference between the marginal entropy (15) and conditional entropy (16)

$$I(X;Y) = H(X) - H(X|Y) \qquad (18)$$

Thus, for a given sequence $x = \{x_0, x_1, x_2, ..., x_n\}$, the *Information Rate* of sequence x is defined

$$IR(x) = \frac{1}{n}\sum_{i=1}^{n} H(x_{0:i}) - H(x_{0:i}|x_{0:i-1}) \qquad (19)$$

where $H(x)$ is the entropy of x, and is estimated based on statistics of the sequence up to event x_i using a unigram language modeling approach. We compute the distribution of symbols in sequence x as the frequency of occurrence over sequence length.

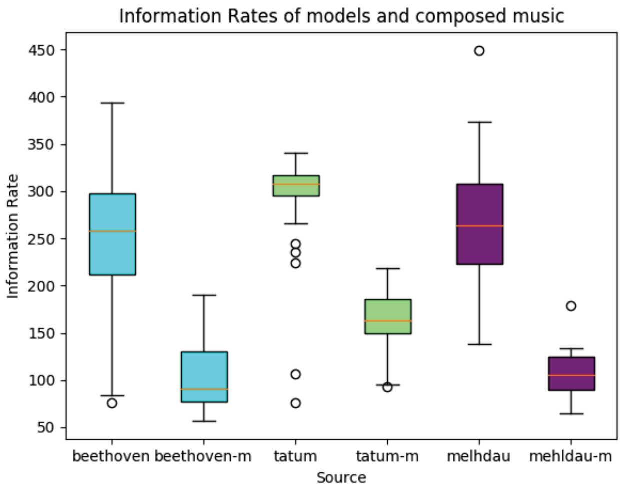


Fig. 4. Information Rate comparison of models with composers. Any annotated with '-m' are generated from models.

Intuitively, larger values of IR occur when repetition and variation are in balance, and smaller values of IR occur when sequences are either random or very repetitive [23]. For example, a high IR corresponds to when specific events occur rarely, but are highly likely given their parents. This often occurs in sequences with higher-level repetitive structure.

Fig 4 contains box plots of Information Rates calculated with a subsample of k randomly selected generated pieces taken from different models for comparison with composed music; namely all 32 Beethoven piano sonatas as performed by Daniel Barenboim, Brad Mehldau's *10 Years Solo* album of live performances, and Art Tatum's *Solo Masterpieces* compiled by Pablo records. Standard deviations are represented by whiskers, means by orange lines inside each box, and outliers by circles. Appendix VI contains a full box plot of information rates across all models.

Qualitative

We also provide intuitive, qualitative measures of model performance on the task of generating realistic musical performances with the intent of capturing self-referential structure. We restrict the length of any two samples, either generated or composed, to a maximum of (30 sec) for consistency and the time commitment of survey respondents. The survey contains 10 pairs of examples of generated audio from our models and are asked to rate their respective musicality and estimated style of a given composer.

We evaluate our models utilizing the Kruskal-Wallis H test and Wilcoxon signed-rank tests for matched pairs with significance value $\alpha = 0.01$. After presenting participants with two examples taken from a subset of the model space, we asked which of the two were more musical on a Likert scale from 1 to 5. We conducted a post-hoc analysis on the comparisons and using Bonferroni correction [27]. 400 ratings were collected with each model involved in 80 pairwise comparisons. Figure 5 shows the number of instances

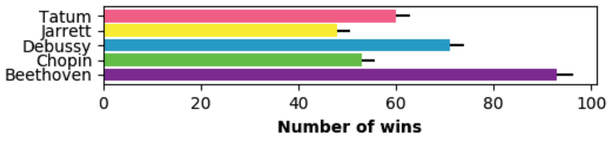


Fig. 5. Listening Test results, displaying number of times a model won in a pairwise comparison with error bars showing estimated standard deviation of means.

a model was chosen as being more musical. Samples from different models can be listened to at the following link:

https://www.cs.rit.edu/∼jps9013/

Participant Segmentation: We asked participants to self-identify as either musical laypersons, amateur musicians, or industry professionals into these categories to infer differences between the populations. Industry professionals performed best on the task of correctly identifying model-composer style, while laypersons scored the least on average. This observation aligned with our expectations that trained musicians would be more familiar with composer style, perform well on identification of style, and seems to suggest model capacity to re-create a particular composer's style is self-consistent. If the models did not accurately represent composer style, we should expect no differences between the groups with random results.

Survey Results: A Kruskal-Wallis H-test of the ratings showed at least one statistically significant difference between the models: $\chi^2(4) = 21.52$, $p < 0.00025$. A post-hoc analysis using the Wilcoxon signed-rank test showed a statistically significant difference between the Chopin and Beethoven models, as well as the Beethoven and Keith Jarrett models, with $p < 0.00025$. Based on the ratings and these results, we found that the Beethoven model is the most aesthetically appealing across all participants. However, within the segments professional musicians overwhelmingly favor the Chopin and Jarrett models. These are counter-intuitive results, as the Beethoven model has the lowest Information Rate score, whereas the Jarrett model has the highest among generative models, with the Chopin model is in the middle of the pack.

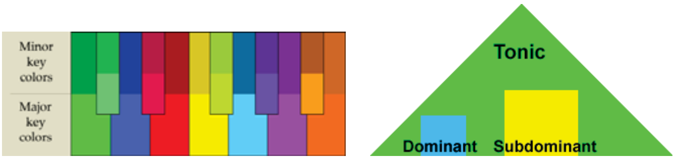
VI. Graphical Results

In addition to the quantitative and qualitative evaluations above, we provide intuitive visual representations of musical examples as a way of informal comparison. We show results from two visualization techniques for audio: *Keyscapes* [28], characterizing tonality, and *Self-Similarity Matrices* [29].

Keyscapes

Tonality is the property of music that music is perceived to be in a specific *key center*, which is to say that there is a hierarchical ordering of pitch class patterns. Perception of tonality is determined by the frequency of occurrence of pitches in the musical piece [30].

We utilize the humdrum keyscape tool by David Huron [28] to parse the tonality of a given MIDI file using the Krumhansl-Schmuckler key-finding algorithm [31]. The results are converted to a PNG image file from the humdrum native data format. The peak of the pyramid shows the key estimation for the entire piece, and moving down toward the base is a recursive set of decreasing window sizes which give an estimation of the key estimates within that context window. Intuitively, the more self-reference that exists in a piece corresponds to a highly structured keyscape, with tonal centers that recur often in fractal patterns and are visibly-discernible to the human eye.



((a)) *Newton* Colormap for keyscapes organized by pitch-classes

((b)) Toy example demonstrating tonal clustering within a keyscape

Fig. 6. Keyscape features

Fig 6 contains a legend to understand tonal clustering in Fig 7, which shows keyscapes of generated pieces from different generative models.

Self-similarity Matrix

A musical piece's *structural organization* communicates a sense of coherence over its duration in which thematic material, motivic patterns, phrases, and whole sections are repeated throughout. Self-similarity matrices encode this property well and make visually inferring patterns corresponding to musical structure a straightforward geometric problem. The entries (i, j) of matrix S express the (dis)similarity between music at positions i and j. This structural coherence is what arguably makes listening to music for humans aesthetically pleasing and interesting [29].

To visualize the structural organization of a musical piece, we construct a self-similarity matrix S along the time dimension by computing the Short Time Fourier Transform of a $1D$ input waveform X with 50% overlapping Hamming

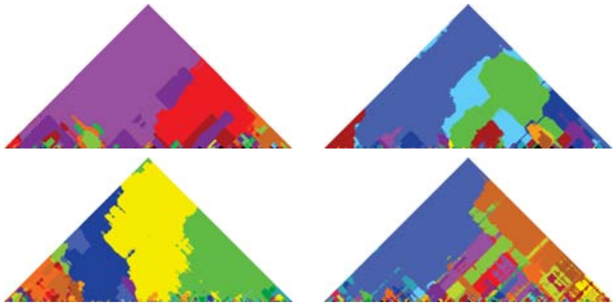


Fig. 7. Keyscapes of generated examples from different models:

Top Left: Beethoven
Top Right: Chopin
Bottom Left: Brad Mehldau
Bottom Right: Art Tatum

windows of N sample lengths, around 10ms. We then derive the magnitude spectrogram V by taking the absolute value of each element and normalizing its columns using the Euclidean norm. Subsequently, we compute the cosine similarity between each feature vector in V (moment in time parameterized by the Hamming window) to obtain S.

Justification for cosine similarity comes from Natural Language Processing techniques such as Word2Vec [32], where the degree of difference between orientations of two word vectors A and B corresponds to the angle θ. Thus, $cos(\theta)$ will be zero when the vectors are orthonormal and linearly independent, indicating no or little similarity. Similarly oriented vectors may have similar meaning, such as in the characterization of semantic relatedness of sentences in a given corpora.

Cosine similarity of two vectors is defined as follows:

$$Sim(A,B) = cos(\theta) = \frac{A \cdot B}{||A||\ ||B||} \tag{20}$$

Thus to obtain S from two spectrograms V^a and V^b:

$$S = Sim(V^a, V^b) = \frac{\sum_{i=1}^{n} V_i^a V_i^b}{\sqrt{\sum_{i=1}^{n} V_i^{a2}}\sqrt{\sum_{i=1}^{n} V_i^{b2}}} \tag{21}$$

where $n =$ no. of frequency channels, in this case. Matrix S will contain a measure of the relatedness between pieces V^a and V^b, symmetric about the diagonal. Appendices II and III contain generated and ground truth similarity matrices.

To compare generated examples, we compute the *mean self-similarity score* MSS by taking the mean of cosine similarity matrix S as a measure of structural self-similarity,

$$\begin{aligned} MSS(S) &= \textit{Mean}(S) \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} S(i,j) \end{aligned}$$

where N is the number of windowed sections (rows of S).

We then sample k generated examples from each model and average the individual MSS within each model subsample, obtaining a measure of the proclivity to output structured content for the i^{th} model on its set of sampled similarity matrices D_i,

$$M_{score}(D_i) = \frac{1}{K}\sum_{k=1}^{K} MSS(S_k) \tag{22}$$

Appendix V contains a table of self-similarity matrix M_{score}s.

VII. Conclusion

According to subjective evaluations from participants, the models produce very pleasing music with good balance between varying and repetitive structure. The generated examples contain significant self-reference up to ∼100 seconds, where most samples wander off toward randomness. They often lack 'complete' musical phrasing and precise harmonic progressions in predictable intervals.

We found that sampling with decode lengths up to twice that which models were originally trained on seem to produce better results, on average. Shorter decode lengths below 2048 typically produced hurried and over-saturated examples with many notes in a short period of time, whereas longer decode lengths tended to produce slower, more laborious development of melodic material. This is suggestive of a kind of time invariance with respect to information content of a given sample, such that the relative entropy contained remains roughly constant with variable decode length. This agrees with the presuppositions of Huang et al. (2018) [9] in that the Transformer architecture may produce coherent examples for sequence lengths much longer than what models were initially trained with.

VIII. Future Work

This paper aims to provide the groundwork for further study of differences in human perception based on varying segmentation criteria on the boundary between the fields of machine learning and artificial intelligence, sonification, and cognitive computing.

Further experimentation with data from different genres is needed. Large unlabeled music corpora can now be explored with varying degrees of specification with the transcription framework. Ensuring homogeneity of new datasets is essential. A logical extension would be the exploration of further isolated genres, such as so-called pop, rock & roll, country, 20th century (also known as serial) music, etc. and their subcategories.

It is also desirable to design a future study which models time period and cultural relativity, as the classical music models described in this paper on spans a time period of over 200 years, from 1703 to 1910. The jazz data sets span a period from approximately 1908 to 2012, over which an incredible amount of innovation and cultural change has taken place which impact the perceptions of lay and trained listeners alike. Thus, fine-grained studies of the relationships between classes, socioeconomic backgrounds, and cultural heritages of listeners is needed.

Additionally, an age-related study that segments population based on age and musical preference is desirable.

Acknowledgment

It is with gratitude that I'd like to thank Dr. Hans-Peter Bischof for supporting this foray into an atypical research project, whose constant guidance has afforded me opportunities I could not have imagined. Thanks to Dr. Richard Zanibbi for many hours of consultation and whose domain expertise helped make this work possible. I would also like to thank Curtis 'Fjord' Hawthorne, Ian Simon, Adam Roberts, and the rest of the Magenta team at Google for providing patient responses throughout our correspondence.

References

[1] “Magenta make music and art using machine learning,” https://magenta.tensorflow.org/, accessed: 2019-01-17.

[2] F. T. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic stylistic composition of bach chorales with deep LSTM,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 449–456. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/156_Paper.pdf

[3] C. A. Huang, T. Cooijmans, A. Roberts, and A. C. C. and.org.

[4] A. Roberts, J. Engel, and D. Eck, Eds., *Hierarchical Variational Autoencoders for Music*, 2017. [Online]. Available: https://nips2017creativity.github.io/doc/Hierarchical_Variational_Autoencoders_for_Music.pdf

[5] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” *CoRR*, vol. abs/1803.02155, 2018. [Online]. Available: http://arxiv.org/abs/1803.02155

[6] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” *CoRR*, vol. abs/1705.03122, 2017. [Online]. Available: http://arxiv.org/abs/1705.03122

[7] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[9] C. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, “An improved relative self-attention mechanism for transformer with application to music generation,” *CoRR*, vol. abs/1809.04281, 2018. [Online]. Available: http://arxiv.org/abs/1809.04281

[10] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, “Neural audio synthesis of musical notes with wavenet autoencoders,” *CoRR*, vol. abs/1704.01279, 2017. [Online]. Available: http://arxiv.org/abs/1704.01279

[11] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, “This time with feeling: Learning expressive musical performance,” *CoRR*, vol. abs/1808.03715, 2018. [Online]. Available: http://arxiv.org/abs/1808.03715

[12] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[13] M. Mller, V. Konz, and M. C. Sebastian, “A multimodal way of experiencing and exploring music.”

[14] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[15] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” *CoRR*, vol. abs/1710.11153, 2017. [Online]. Available: http://arxiv.org/abs/1710.11153

[16] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” *CoRR*, vol. abs/1601.06759, 2016. [Online]. Available: http://arxiv.org/abs/1601.06759

[17] L. Theis and M. Bethge, “Generative image modeling using spatial lstms,” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1927–1935. [Online]. Available: http://papers.nips.cc/paper/5637-generative-image-modeling-using-spatial-lstms.pdf

[18] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications,” *CoRR*, vol. abs/1701.05517, 2017. [Online]. Available: http://arxiv.org/abs/1701.05517

[19] S. Hochreiter, Yoshua, F. F. Informatik, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.”

[20] L. J. Ba, R. Kiros, and G. E. Hinton, “Layer normalization,” *CoRR*, vol. abs/1607.06450, 2016. [Online]. Available: http://arxiv.org/abs/1607.06450

[21] R. Józefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, “Exploring the limits of language modeling,” *CoRR*, vol. abs/1602.02410, 2016. [Online]. Available: http://arxiv.org/abs/1602.02410

[22] “Protocol Buffers google description,” https://developers.google.com/protocol-buffers/, accessed: 2010-02-28.

[23] S. Lattner, M. Grachten, and G. Widmer, “Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints,” *CoRR*, vol. abs/1612.04742, 2016. [Online]. Available: http://arxiv.org/abs/1612.04742

[24] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[25] R. Durrett, *Probability: Theory and Examples*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010. [Online]. Available: https://books.google.com/books?id=evbGTPhuvSoC

[26] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2012. [Online]. Available: https://books.google.com/books?id=VWq5GG6ycxMC

[27] J. Dunn and O. J. Dunn, “Multiple comparisons among means,” *American Statistical Association*, pp. 52–64, 1961.

[28] D. Huron, “Music information processing using the humdrum toolkit: Concepts, examples, and lessons,” *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002. [Online]. Available: https://doi.org/10.1162/014892602760137158

[29] J. Foote and M. Cooper, “Visualizing musical structure and rhythm via self-similarity,” *FX Palo Alto Laboratory, Inc.*, 01 2001. [Online]. Available: http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/FXPAL-PR-01-152.pdf

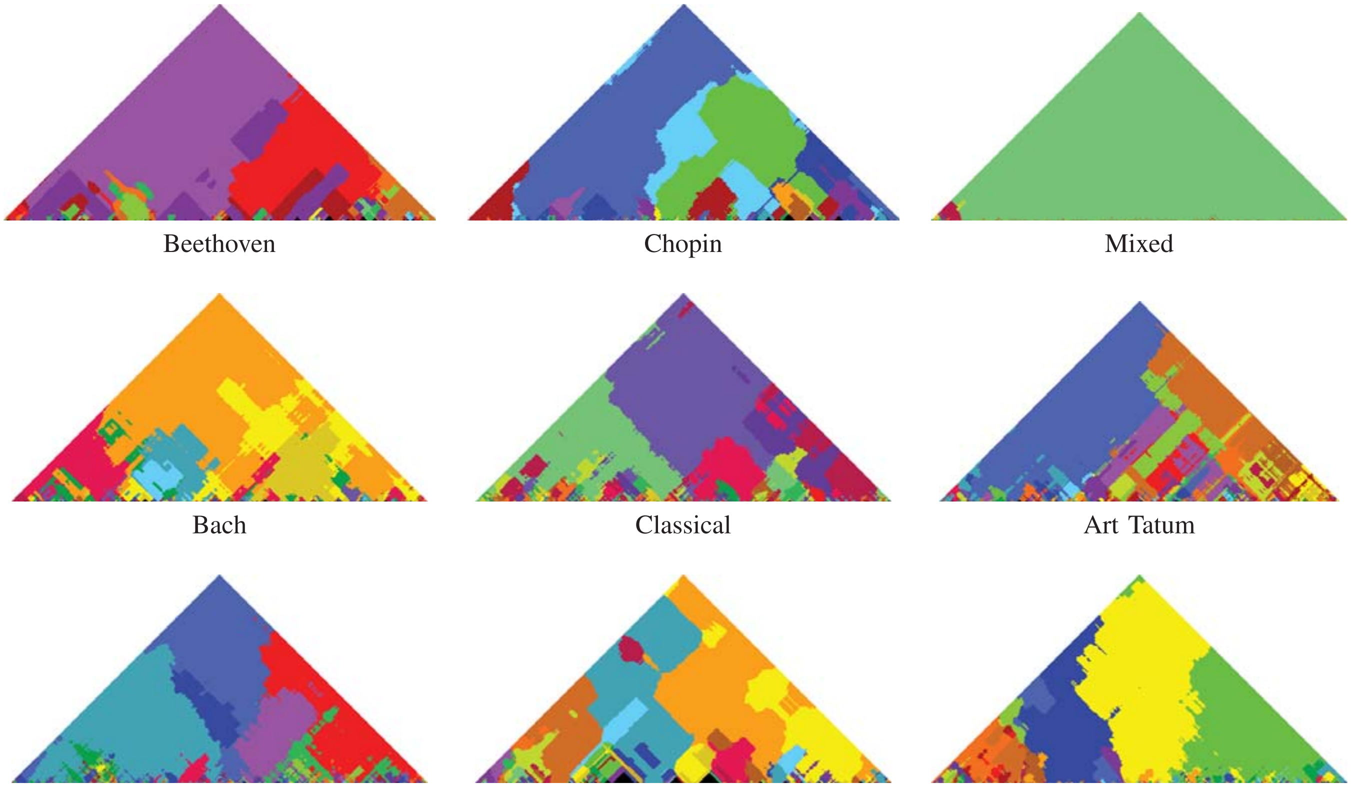
[30] N. Smith and M. Schmuckler, “The perception of tonal structure through the differentiation and organization of pitches,” *Journal of experimental psychology. Human perception and performance*, vol. 30, pp. 268–86, 05 2004.

[31] D. Temperley, “What’s key for key? the krumhansl-schmuckler key-finding algorithm reconsidered,” *Music Perception: An Interdisciplinary Journal*, vol. 17, no. 1, pp. 65–100, 1999. [Online]. Available: http://mp.ucpress.edu/content/17/1/65

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

APPENDIX

I. KEYSCAPES



Beethoven Chopin Mixed

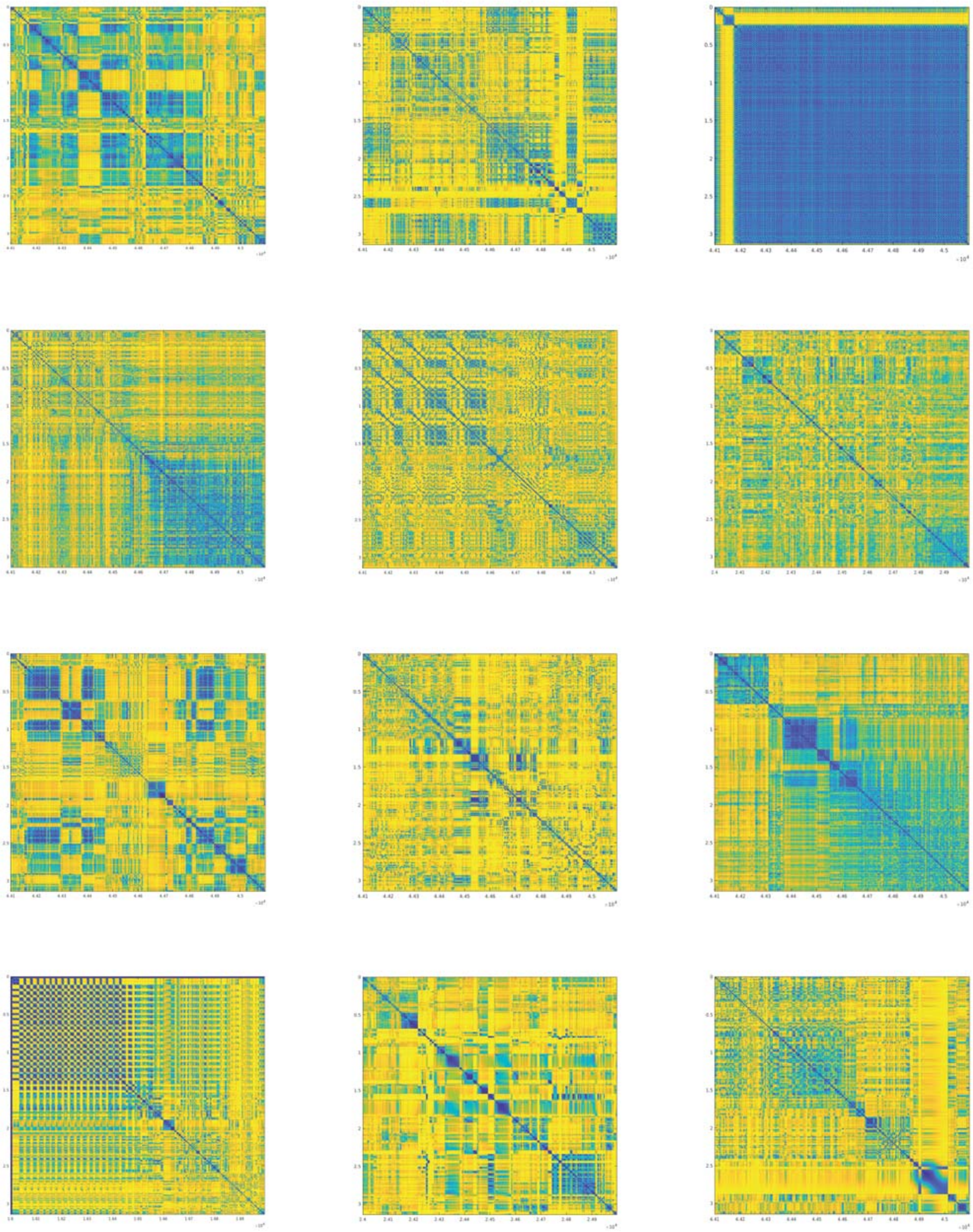
Bach Classical Art Tatum

Maestro Liszt Brad Mehldau

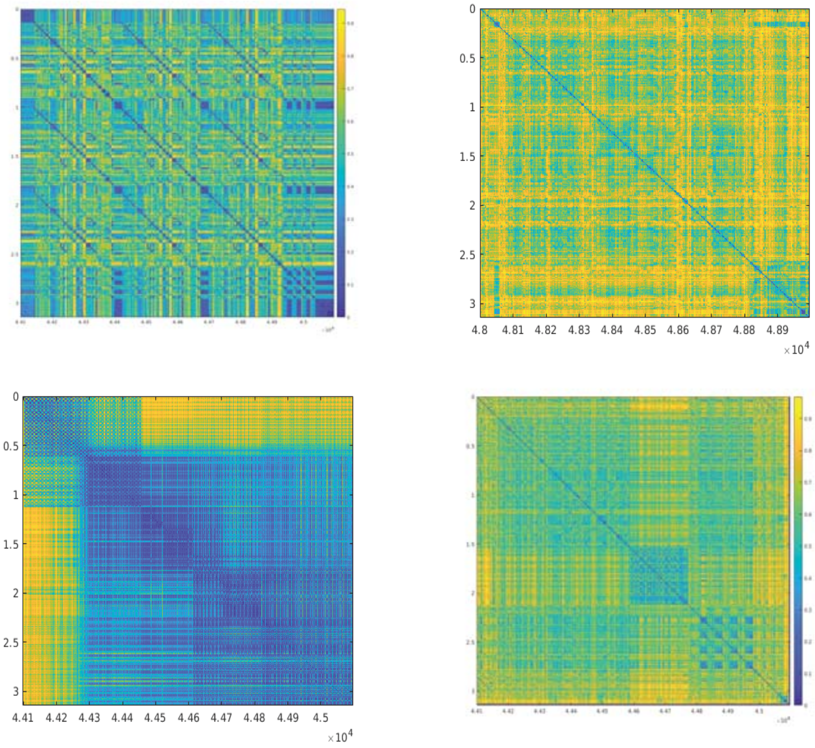
All of the examples above are generated by different models. We include one ”failure” example for illustration at top right, sampled from the mixed genre model. Note it is largely uniform, with most of the area colored lime-green, indicating that the piece is in one key a majority of its duration. The associated similarity matrix is shown below for the same example in the top right of the table below.

II. Self-Similarity Matrices of Model Generated Music

The table below contains examples of self-similarity matrices of samples taken from models shown above in the keyscape table (Appendix I), in the same configuration for ease of comparison.



III. Self-Similarity Matrices of Human Composed Music



Self-Similarity Matrices of human composed pieces:
Top left: Brad Mehldau's *John Boy*
Top right: Brahms *Symphony no. 2 Allegro non troppo*
Bottom left: Dawn of Midi's *IO*
Bottom right: Chick Corea's *Matrix*

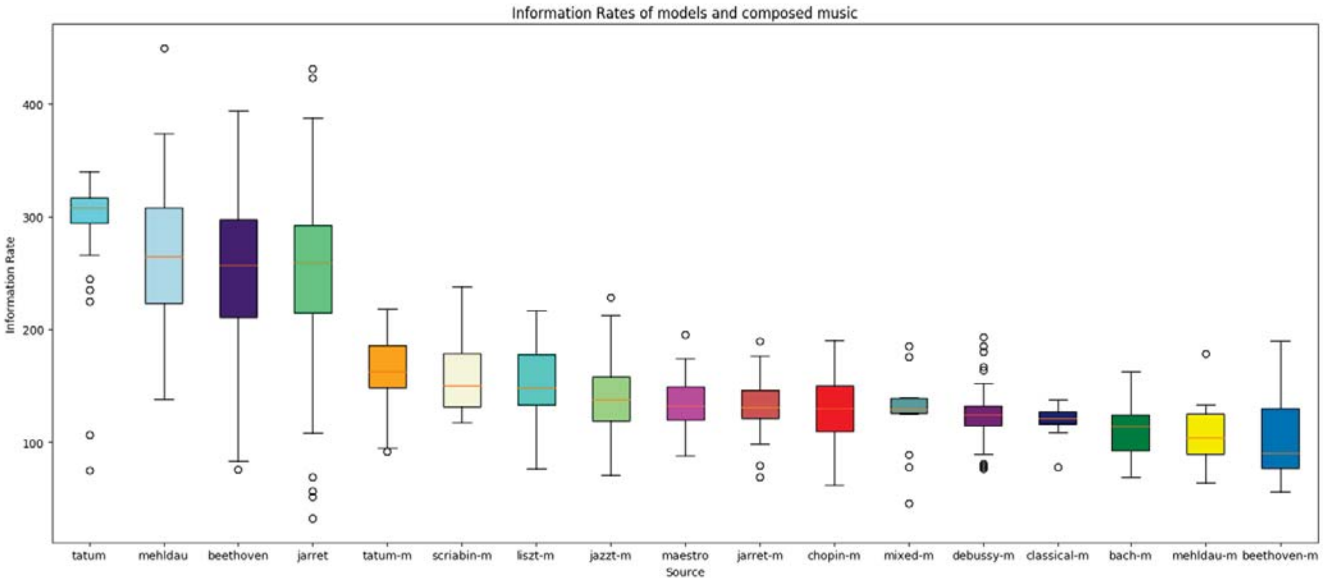
IV. Quantitative Results

Model	NLL	NLP	IR
Bach	1.91	-1.91	112.03
Brahms	1.76	-1.75	161.44
Beethoven	1.79	-1.792	105.62
Keith Jarrett	1.65	-1.66	131.38
Brad Mehldau	1.26	-1.26	155.01
Chopin	2.17	-2.17	128.01
Debussy	1.99	-1.98	125.74
Liszt	1.79	-1.78	155.01
Maestro	1.82	-1.81	135.39
Scriabin	2.19	-2.21	157.59
Tatum	2.82	-2.73	106.7
Classical	1.75	-1.74	151.38
Jazz-full	1.95	-1.95	169.12
Mixed	1.75	-1.75	126.06

V. Mean Self-Similarity Scores

Model	Mean Self-Similarity Score
Bach	0.6852
Beethoven	0.6979
Brahms	0.6349
Chopin	0.6564
Debussy	0.7227
Keith Jarrett	0.6397
Brad Mehldau	0.5379
Liszt	0.7016
Maestro	0.7376
Scriabin	0.7237
Tatum	0.7101
Classical	0.7495
Jazz	0.6934
Mixed	0.7295

VI. Information Rates


Information Rates of models and composed music
Information Rate
tatum
mehldau
beethoven
jarret
tatum-m
scriabin-m
liszt-m
jazzt-m
maestro
jarret-m
chopin-m
mixed-m
debussy-m
classical-m
bach-m
mehldau-m
beethoven-m
Source