Exploring Psychoacoustic Representations for Machine Learning Music Generation

Bryan Wilson, James Skripchuk, John Bacher
North Carolina State University
Raleigh, North Carolina, USA
{*bcwilso7 ,jmskripc, jtbacher*}*@ncsu.edu*

Abstract

Deep learning music generation systems have made progress in generating music artifacts ranging from scores to audio. The most successful deep learning methodologies require large amounts of computational resources, usually only available to large organizations. The environmental impact of training is non-negligible, and the computational resources can be prohibitive for research groups or independent artists engaging in co-creative design. While successful, many of these models do not take into account existing musicological domain knowledge which could yield better model performance. As a proof of concept, we augment a deep learning music generation model with an extension of a mathematical model of dissonance perception, using it to construct harmonic tension curves as an internal representation in a deep learning model. We train embeddings based on our representation and substitute them in an off-the-shelf transformer music generator. Our representation performs marginally better than baseline, with a significant reduction in training time. We explore how our representation may yield greater control of the generative space. We discuss how these results inform future research in utilizing existing domain knowledge in audio and music in order to augment deep learning models, and suggest pathways for further collaboration between computational creativity and deep learning spaces.

Introduction

Large machine learning models, particularly deep learning models that utilize many parameters to train on and generalize to a large set of data, have demonstrated incredible ability at completing complex tasks with no obvious algorithmic method. Training these large machine learning models however are computationally intensive, taking hours if not days to train. This results in high power consumption, high financial costs, and negative environmental impacts (Strubell, Ganesh, and McCallum 2020). For example, BERT (Bidirectional Encoder Representations from Transformers), a large machine learning model released by Google consisting of 110M parameters, is estimated to require 79 hours for initial training. During this initial training, BERT is estimated to consume 12kW of power, cost between $3,751-12{,}570$, and emit 719 lbs of CO2. (Strubell, Ganesh, and McCallum 2020). This problem affects many domains, including machine learning music generation as models such as MuseNet and Coconet consist of thousands if not millions of parameters. The size and required resources for these large models make it hard for computational creativity researchers to work with them.

Data representation has a significant impact on model training as well as the quality of music generated from machine learning models (Briot, Hadjeres, and Pachet 2017). Symbolic music representations, such as sheet music, MIDI, and chord symbols are known to decrease training time when compared to pure audio representations. Historically, work has been done on constructing additional models of how humans perceive audio and music. However, there is a lack of research on incorporating these existing theories of music perception into current representations of music, which can be useful for improving training efficiency. We aim to address this in our work.

We chose harmonic tension as our representation because it has been considered by a number of music theorists to be a strong indicator of musical coherence (Bigand, Parncutt, and Lerdahl 1996). In addition, many methods have been developed for both quantifying and modeling the harmonic tension and resolution across a piece. Specifically, we focus on the concept of a *tension curve*, a graphical model of the harmonic tension over a given chord progression (Yoo and Lee 2006). Currently, such methods are either limited to western models of music theory or only consider a finite number of chordal tones. Thus, we've designed a novel method of calculating tension curves based on psychoacoustics. To evaluate the impact of this method on the training time of machine learning music generation, we conducted a comparative study on our dataset of tension curves and a symbolic representation of music.

Data representation choice has another advantage, particularly in providing control over the generative space of the ML model. By using a representation that is suited for music similarity, for example, it is possible to take an ML music generation model, which is often seen as a black box (Castelvecchi 2016) and allow the user more control over the generative space. We perform exploratory analysis on the models output to determine the potential for greater harmonic controllability.

Related Work

Methods of Improving Training Efficiency

Current methods of improving training efficiency either fall within framework level optimization, parallel opportunities, or hardware developments (Sharir, Peleg, and Shoham 2020). Framework level optimization such as regularization and adaptive learning rate have been commonly utilized for improving model performance and training efficiency (Staib et al. 2019). However, more complex optimization approaches, such as co-designed algorithms and natural gradients have emerged more recently. Though these algorithms can lead to a quicker training time, they also can result in worse model performance (Wang et al. 2022). Current parallel opportunities, mainly within Distributed ML, are divided into two categories: data parallelism and model parallelism (Wang et al. 2022). Data parallelism requires the data to be partitioned between different nodes before fed into multiple instances of the machine learning model for training. Model parallelism requires the machine learning model be split up and placed on different devices in such a way that it can still be trained concurrently (Peteiro-Barral and Guijarro-Berdiñas 2013). While distributed ML has demonstrated success in improving training efficiency, it is very difficult to implement and more vulnerable to system failure as components are decentralized (Peteiro-Barral and Guijarro-Berdiñas 2013). Computational efficiency at the hardware level has also shown promise in improving training efficiency. There are many hardware development approaches such as memory management, dedicated hardware, and resource allocation (Markidis et al. 2018). Such approaches however have physical limitations that require constant iterations as machine learning model size increases.

Tension Curves

Even though there have been developments in expert authored music representations (Downie 2003), they haven't been utilized for machine learning music generation. One of the most notable of these representations is harmonic tension curves (Sethares 1993; Plomp and Levelt 1965; Navarro-Cáceres et al. 2020a; Yoo and Lee 2006). A harmonic tension curve models the harmonic tension and resolutions over a given piece of music by mapping a combination of tones within a chord into a single value. Common approaches are geometric mappings based on the distances between notes, such as Lerdahl's Tonal Pitch Space (Lerdahl and others 2001) or Chew's Spiral Array (Chew 2000). While useful, these do not capture any information about how humans physically perceive dissonance.

Krumhansl (Krumhansl and Shepard 1979) constructed a method where subjects were to assign a numerical rating of stability of certain pitches within a scale. While this approach takes into account human perception, it can only calculate the dissonance values of twelve notes in respect to a certain scale, and doesn't take into account the full complexity of the interaction of a note and its overtones. To mitigate this, we construct a mapping function based on an existing mathematical model of the perceived dissonance between two or more notes. To do this, we build on the approach of Vassilakis (Vassilakis and Fitz 2007), who parameterized a dissonance curve derived by Plomp and Levelt (Plomp and Levelt 1965). Not only does this allow the calculation of a dissonance value for any arbitrary collection of notes no matter the tuning or temperament, but it also includes the interaction between any arbitrary notes and their overtones.

Tension Calculation

Dissonance for Three or More Tones

In this section, we build on the work of Vassilakis to formulate a tension function able to consider a chord with an arbitrary size within the context of a piece. First, we expand Vassilakis's dissonance function to consider a chord of an arbitrary size. For chords with more than two complex tones, we calculate the dissonance of every combination of complex tones. We define D as the dissonance function developed by Vassilakis. The resulting dissonance function then becomes

$$D_v(C) = D(C_1, C_2, \ldots, C_n) = \sum_{i=1}^{Nc} D(C_a, C_b)$$

where C_a and C_b is a unique complex tone combination from the set $\{C_1, C_2, \ldots, C_n\}$ and N_c is the number of possible complex tone pairs within C.

Harmonic Tension Calculation

We will now introduce contextual components. In addition to vertical dissonance, we will also consider key tonal distance and contextual tension, as inspired by (Navarro-Cáceres et al. 2020b). However rather than use different models to calculate each component, we will be using the same dissonance function.

In regards to key tonal distance, we represent the key of our piece as a chord where each note in the key is represented in the scale. We will represent a chord representation of a key with a K where $K = [K_1, K_2, \ldots, K_n]$. Given a chord L, we will superimpose the notes of L onto the notes of K making sure to remove all duplicate notes. The dissonance therefore is calculated as

$$D_k(L, K) = D\{L_1, L_2, \ldots, L_n, K_1, K_2, \ldots, K_n\}$$

Contextual tension is based on the understanding that the perception of a chord is influenced by the chords that precede it. Similar to finding key tonal distance, we will superimpose the chord of interest onto the chord before making sure to eliminate any duplicates. Given two chords M and N with notes $[M_1, M_2, \ldots, M_n]$ and $[N_1, N_2, \ldots, N_n]$ respectively. The dissonance therefore is calculated as

$$D_P(M, N) = D\{M_1, N_2, \ldots, N_n, M_1, M_2, \ldots, M_n\}$$

Now that we have defined how to calculate every component of tension we will consider in this paper, we will now define how we aggregate these components to calculate total tension. Suppose we have a chord C_n where n is the chord position in a given piece of music in the key of K. Then we will define the total tension of chord C_n as

$$D_T(C_n) = D_v(C_n) + D_k(C_n, K) + \sum_{i=1}^{W} \gamma^i D(C_n, C_{n-i})$$

for $i \geq n$ where W is the window size and γ is the decay. Window size, W, determines how many chords before the chord of interest we consider in our contextual tension calculation. Decay, γ, determines how much our contextual tension is influenced by chords further in the past. These two values will serve as parameters to control for how much a chord's previous context influences its tension value.

Methodology

Data and Preprocessing

Our data consisted of 329 Bach chorales provided from the Music21 library (Cuthbert and Ariza 2010) designed for music analysis and processing. For every chorale in our dataset, we extracted the chords placed on the strong beats and transformed them into a list of vectors. For our tension representation, we applied our tension function to our dataset of chord vectors. For the window size parameter, W, we chose the values 1, 2, 3, 4, 5, 6, and 7 due to Bach's typical 8 beat phrasing. Since the decay parameter is confined to the range [0, 1], we chose 0.125, 0.25, 0.5, 0.75 and 0.875, in order to have an equidistant spread of values across its range. We passed our vector dataset into our tension function for all combinations of W and γ, resulting in 35 datasets of tension values. For each dataset, we allocated 80% for training, 10% for validation, and 10% for testing. Figure 1 shows a diagram of the pipeline followed for our experiment.

Training Procedure

We perform a comparative study on our symbolic representation as ground truth and our tension representation. We utilized the Music Transformer model developed by Huang et al. due to its recency in development and its manageable overhead (i.e. required training time, training data, computation power, etc) compared to other Transformer models (Huang et al. 2018). We trained our Music Transformer on the symbolic dataset using its given embedding layer and on each of our tension datasets replacing the existing embedding layer with our pretrained embedding layer. We used a batch size of 64 and trained our model for 50 epochs, each epoch consisting of 155 iterations to ensure our Music Transformer did not overfit. We used Cross Entropy Loss to evaluate the loss for each model prediction. For accuracy, we averaged the number of correct predictions across a chorale.

Results

Model Performance Analysis

We first look at the training and validation loss and accuracy curves acquired after training our Music Transformer on the symbolic dataset and our tension datasets. We only include one graph, Figure 2, for space concerns, however it depicts the benefits of our representation at a high level. Our tension representations starts at a higher training and validation loss but results in a lower training and validation loss compared to the symbolic representation. Similarly, our tension representations results in a lower training and validation accuracy but results in a higher training and validation accuracy for both compared to the symbolic representation. In addition, our tension representation converges to a lower training and validation loss and a higher training and validation accuracy quicker then the symbolic representation.

Test Dataset	Loss	Accuracy
Symbolic Control	0.671	0.823
$\gamma = 0.125$	0.667	**0.824**
$\gamma = 0.75$	**0.613**	0.818

Table 1: Best performing parameters for $W = 7$

Test Dataset	Loss	Accuracy
Symbolic Control	0.671	0.823
$W = 4$	**0.666**	0.822
$W = 6$	0.670	**0.823**

Table 2: Best performing parameters for $\gamma = 0.5$

In regards to window size, higher window sizes start with higher training and validation loss values and lower training and validation accuracy values, but end with lower training and validation loss values and higher training and validation accuracy values compared to the symbolic representation. Additionally, higher window sizes increase the rate of convergence for all loss and accuracy curves. These effects begin to diminish however for window sizes greater than 3. Decay however, had no significant effect on initial and ending values for training and validation loss and accuracy curves as well as their rates of convergence.

To evaluate the influence of our tension representation on our Music Transformer's ability to generalize to new data, we compared the testing loss and accuracy values obtained from our tension representation to that of the symbolic representation. The dataset with $\gamma = 0.75$ and $W = 7$ produced a lower testing loss compared to the symbolic representation and the dataset with $\gamma = 0.125$ and $W = 7$ produced a higher accuracy compared to that of our symbolic representation. Nevertheless, there are no significant improvements in testing loss and accuracy using our tension representation compared to our symbolic representation. Table 1 and Table 2 show testing loss and accuracy across tension function parameters $W = 7$ and $\gamma = 0.5$ respectively.

Discussion: Overall, our tension representation yields better training and validation loss and accuracy values as well as a quicker convergence time compared to the symbolic representation. This suggests that having a representation informed by human perception may allow for faster ML model training. Furthermore, using some form of intermediate representation, such as ours, to reduce training time would be beneficial for those looking to generate music with

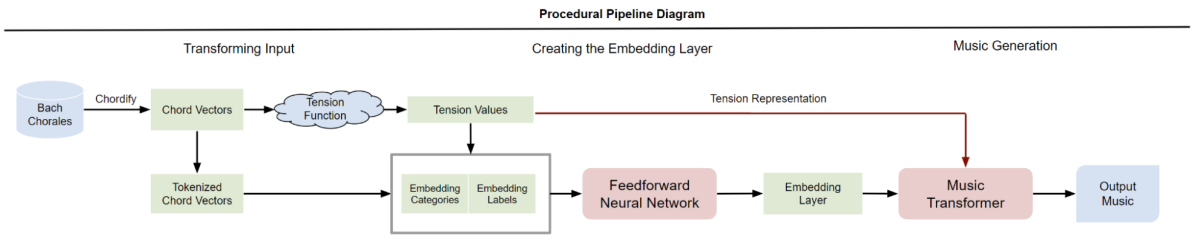

Figure 1: General Procedural Pipeline for Music Generation

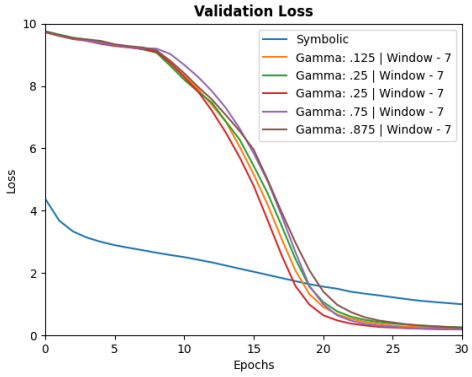

Figure 2: Validation Loss: For all γ and W = 7

limited resources.

Model Output Results

We explore our Music Transformer's output using an accepted metric of harmonic variation, Chord Histogram Entropy (CHE), proposed by (Yeh et al. 2021). To observe any correlations our decay parameter has with harmonic variation, we set $W = 7$ and calculated the CHE of both our tension representation outputs for all decay values. Our calculations exhibit a parabolic correlation, with R^2 = .875, between decay and CHE. There is not a clear correlation however, between our window size parameter and CHE.

Discussion: Our results suggest that the decay parameter, γ, has a parabolic correlation with harmonic variation. Even though we are only able to establish correlation, these results leave room for future work to determine if a causal relationship exists. Nevertheless, our model demonstrates the potential for more control and creativity focused ML models that rely on existing knowledge rather than brute-force generation. There is clearly more work to be done on making models that are sufficient for music generation tasks without the overhead of long training time and resource consumption.

Threats to Validity and Future Work

In this work, we expanded the dissonance function, proposed by Vassilakis, to incorporate both an arbitrary number of chordal tones and contextual information such as key and previous chords which we then utilized to generate a dataset of tension values to train our Music Transformer model on. Our results on model performance suggests that incorporating human perception into ML training results in higher accuracy, lower loss, and quicker training time all while producing comparable testing results. Furthermore, our results on model output explore the relationship between our tension representation parameters and the harmonic characteristics of our Music Transformer's output suggesting a correlation between contextual harmonic information and harmonic variance.

One limitation is the absence of subjective evaluation metrics such as a case study or a listening test. This makes it difficult for us to create any strong claims on the influence of our tension representation on music quality. Another limitation is that by only extracting chords on the strong beats, we limit the chord voicing range and rhythmic variance of our generated music, making it unusable for practical applications. Due to the lack of clearly detailed objective music evaluative methods, we only explored one aspects of harmonic structure, leaving many harmonic characteristics of our generated output unexplored. However due to the lack of research in utilizing psychoacoustic models for ML music generation, we believe that our limitations are valid and will be helpful for further studies in this area.

In addition to the future work that can be made to mitigate the limitations mentioned above, we used window size and decay as tension function parameters to control influence of previous chords on tension value. Our research suggests that window size influences loss and accuracy initial and ending values as well as convergence time. *What other parameters can be included in our tension function to further improve model training speed?* Furthermore, our research suggests decay exhibits a correlation to harmonic variance. However, *does this parameter influence harmonic variance?* And if so, *what other parameters can be included in a tension function to influence other music characteristics?* In addition, work has been done in performing tension curve alterations using geometric formulas to reharmonize a chord progression (Yoo and Lee 2006). *In what ways can we utilize the geometric transformation of tension curves to control harmonic interest in model output?* Finally, we only considered modeling harmony for computational music generation. *What other models can we create to influence machine learning training efficiency such as rhythm, melody, and texture through computational music generation?*

References

Bigand, E.; Parncutt, R.; and Lerdahl, F. 1996. Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception & psychophysics* 58:125–141.

Briot, J.-P.; Hadjeres, G.; and Pachet, F.-D. 2017. Deep learning techniques for music generation – a survey.

Castelvecchi, D. 2016. Can we open the black box of ai? *Nature News* 538(7623):20.

Chew, E. 2000. *Towards a mathematical model of tonality*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Cuthbert, M. S., and Ariza, C. 2010. music21: A toolkit for computer-aided musicology and symbolic music data.

Downie, J. S. 2003. Music information retrieval. *Annual review of information science and technology* 37(1):295–340.

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2018. Music transformer.

Krumhansl, C. L., and Shepard, R. N. 1979. Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of experimental psychology: Human Perception and Performance* 5(4):579.

Lerdahl, F., et al. 2001. *Tonal pitch space*. Oxford University Press, USA.

Markidis, S.; Chien, S. W. D.; Laure, E.; Peng, I. B.; and Vetter, J. S. 2018. NVIDIA tensor core programmability, performance & precision. *CoRR* abs/1803.04014.

Navarro-Cáceres, M.; Caetano, M.; Bernardes, G.; Sánchez-Barba, M.; and Merchán Sánchez-Jara, J. 2020a. A computational model of tonal tension profile of chord progressions in the tonal interval space. *Entropy* 22(11):1291.

Navarro-Cáceres, M.; Caetano, M.; Bernardes, G.; Sánchez-Barba, M.; and Merchán Sánchez-Jara, J. 2020b. A computational model of tonal tension profile of chord progressions in the tonal interval space. *Entropy* 22(11):1291.

Peteiro-Barral, D., and Guijarro-Berdiñas, B. 2013. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence* 2:1–11.

Plomp, R., and Levelt, W. J. M. 1965. Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America* 38(4):548–560.

Sethares, W. A. 1993. Local consonance and the relationship between timbre and scale. *The Journal of the Acoustical Society of America* 94(3):1218–1228.

Sharir, O.; Peleg, B.; and Shoham, Y. 2020. The cost of training NLP models: A concise overview. *CoRR* abs/2004.08900.

Staib, M.; Reddi, S. J.; Kale, S.; Kumar, S.; and Sra, S. 2019. Escaping saddle points with adaptive gradient methods. *CoRR* abs/1901.09149.

Strubell, E.; Ganesh, A.; and McCallum, A. 2020. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 13693–13696.

Vassilakis, P. N., and Fitz, K. 2007. Sra: A web-based research tool for spectral and roughness analysis of sound signals. In *Proceedings of the 4th Sound and Music Computing (SMC) Conference*, 319–325.

Wang, H.; Qu, Z.; Zhou, Q.; Zhang, H.; Luo, B.; Xu, W.; Guo, S.; and Li, R. 2022. A comprehensive survey on training acceleration for large machine learning models in iot. *IEEE Internet of Things Journal* 9(2):939–963.

Yeh, Y.-C.; Hsiao, W.-Y.; Fukayama, S.; Kitahara, T.; Genchel, B.; Liu, H.-M.; Dong, H.-W.; Chen, Y.; Leong, T.; and Yang, Y.-H. 2021. Automatic melody harmonization with triad chords: A comparative study. *Journal of New Music Research* 50(1):37–51.

Yoo, M.-J., and Lee, I.-K. 2006. Musical tension curves and its applications. In *ICMC*. Citeseer.