2017

# Towards a Relative-Pitch Neural Network System for Chorale Composition and Harmonization

Samuel P. Goree
*Oberlin College*

# Towards a Relative-Pitch Neural Network System for Chorale Composition and Harmonization

Sam Goree
Advisors: Benjamin Kuperman and Adam Eck

March 26, 2017

**Abstract**

Computational creativity researchers interested in applying machine learning to computer composition often use the music of J.S. Bach to train their systems. Working with Bach, though, requires grappling with the conventions of tonal music, which can be difficult for computer systems to learn. In this paper, we propose and implement an alternate approach to composition and harmonization of chorales based on pitch-relative note encodings to avoid tonality altogether. We then evaluate our approach using a survey and expert analysis, and find that *pitch-relative* encodings do not significantly affect human-comparability, likability or creativity. However, an extension of this model that better addresses the criteria survey participants used to evaluate the music, such as instrument timbre and harmonic dissonance, still shows promise.

## 1 Introduction

### 1.1 Computational Creativity

Computational creativity is an emergent subdiscipline within artificial intelligence research that identifies creativity as a normal facet of intelligence and seeks to replicate it algorithmically. Boden [14] defines creativity as the ability to generate novel and valuable ideas, and identifies two categories of novel ideas: P-creative, or new to the person who generated, and H-creative, or historically unique. She also identifies three ways by which creative ideas occur: combinational creativity, where the creator combines two or more existing ideas into a novel one; exploratory creativity, where the creator moves through an idea space defined by certain constraints and explores the limits of those constraints; and transformation creativity, where the creator changes some constraint of an idea space. While all three methods lead to valuable and novel ideas (for example, much of Shakespeare's work can be identified as the result of combinational creativity), it has historically been easier to model combinational creativity and exploratory creativity using computer-based systems.

Creativity research has also historically proven difficult to evaluate, since creativity is difficult to quantify. Jordanous [16] sets out a fairly comprehensive analysis of evaluation metrics in the literature and recommends, among other observations, that creativity research ought to be evaluated both by quantitative and qualitative metrics, and should be evaluated by independent human evaluators, preferably with expert knowledge in the problem domain, in as transparent a manner as possible. She gives the example study of a questionnaire given to several judges to comparatively evaluate four jazz improvisation systems on a number of attributes of creative music.

## 1.2 Algorithmic Music Composition & Bach Chorale Harmonization

Computer algorithmic music composition is a well-established case study for composers and computational creativity researchers alike. Its origins date to Lejaren Hiller's 1957 *Illiac Suite* for string quartet, which uses both hand-crafted decision rules and Markov chains to choose notes to compose, demonstrating that computer systems could use pseudorandom number generators to make musical decisions.

While several composers, inspired by Hiller, investigate stochastic methods in composition during the next few decades (John Cage, James Tenney and Iannis Xenakis come to mind), developments all involved decision rules and Markov models. Cope [1] starts the modern conversation surrounding algorithmic composition with his Experiments in Musical Intelligence project that use augmented transition networks (an automaton approach to parsing/generating natural language) to generate music in the style of Mozart. Hild *et al.* [2] take a different approach and propose HARMONET, a system for harmonizing [1] Bach chorales using an artificial neural network system to assign chords to the notes of a melody, translate those chords into notes and ornament those notes appropriately.

The use of Bach Chorales as a dataset makes sense. Bach wrote 438 chorales that are all quite short (most are less than a minute of music) and are homogeneous in texture and instrumentation (each is written for chorus split into four voices: soprano, alto, tenor and bass). Bach is so consistent in his use of harmonization rules; these pieces are used in introductory western classical music theory courses around the world to teach both composition and analysis.

The start of machine composition required a great deal of expert knowledge. Two researchers decide to take rule-based approaches to harmonization of Bach chorales. Spangler [5] approaches harmonization through a rule-based method that extracts decision rules from a corpus and stochastically applies them to harmonize new chorales. Phon-Amnuaisuk [9] takes a similar approach and uses constraint-based search techniques to find harmonizations that do not violate a set of rules, however those rules are dictated by an expert and not learned from example data. Depending on expert knowledge makes it difficult to adapt a model to new data, since you need to find yourself a new human expert, and heavily subjective to that expert's taste.

Three later experiments [7, 10, 11] reduce the demand for expert oversight and introduce hidden Markov models to Bach chorale harmonization. However, they take on the assump-

[1] Harmonization, though not of much interest to contemporary composers, is a far more clearly-defined problem domain than composition. It requires an AI system to give accompaniment to a melody by choosing the notes to play at the same time as that melody. Since harmonization is an assignment often given to students of music theory, there is a long history of literature on how to correctly harmonize in order to be consistent with 18th century conventions. The chorales of Johann Sebastian Bach, composed during the first half of the 18th century, provide a large corpus of harmonizations with a fairly homogenous style.

tion that a given note's harmonization probability is only defined in relation to some implied state, which ignores much of the intentionality and relationships over time in music composition that may not be captured in the state space.

## 1.3 Application of Neural Networks to Music Composition

The first researcher to apply neural networks to music composition, Mozer [3], trains a feedforward network to output predictions of the next note of a piece given the previous four notes. This method resembles a 4-gram model for linguistics, but learns the probabilities using backpropagation. He trains his system using ten hand-selected melodies from Bach pieces, mostly minuets.

In contrast to feedforward networks, recurrent neural networks (RNNs)(neural networks with layers that take input both from the previous layer and from themselves on the previous input) are more commonly applied when modeling sequence data, such as music. In 1997, Hochreiter and Schmidhuber [4] introduce the long-short term memory (LSTM) layer as an improvement to RNNs by adding internal memory to each layer. They demonstrate that LSTM layers work much better than competing architectures at modeling dependencies over large distances in time, which has made LSTM RNNs a topic of interest in machine composition and harmonization.

Eck and Schmidhuber [8] apply Schmidhuber's work on LSTM networks to music, working with a corpus of improvisations over a twelve bar blues chord progression. They train the network to predict the pitch of the next note in each improvisation, and use a softmax function to ensure that the output probabilities sum to 1 over all the possible notes. Despite making qualitative claims about the quality of their music, no clear evaluation metrics were used.

Boulanger-Lewandowski *et al.* [15] also make use of RNNs, but introduce them as part of a hybrid model including the structure of restricted Boltzmann machines. They use a variety of models to predict several datasets including Bach chorales and compare them on the log likelihood of predicting the polyphony accurately.

Hadjeres and Pachet [21] introduce the current state-of-the-art model, DeepBach, published after the research this paper describes took place.[2] DeepBach makes use of a deep neural network model that predicts individual pitches through a merging of outputs from LSTM predictor networks, one trained to predict Bach chorales from start to end and one trained to predict chorales from end to start. As a result of the bidirectional method, they cannot generate pieces in order and instead have to generate pieces by Gibbs sampling, which allows them to both compose chorales from scratch and harmonize existing melodies. They evaluate using a combination of perception tests with surveyed groups of both musicians and non-musicians and algorithmic plagiarism analysis.

[2] This paper completed very similar research to ours but did not publish their findings until December 2016, after work on this project had started. Rather than abandon our progress and try to improve upon their methods, we (in the words of Ben Kuperman) "take it as validation of [our] ideas and continue to proceed with [our] project."

## 1.4 Tonality

Any system that wants to compose or harmonize based on Bach or other common-practice era composers needs to grapple with the conventions of tonality, [3] which provide a set of common assumptions and expectations for music that color our interpretation but do not clearly present themselves in musical data. Early researchers transpose all of their training data to a common key [2, 8, 10], transpose each example into every possible key [21] or only select pieces from a single key [3].

Allan [11] notably takes a tonality-agnostic approach where notes are only quantified in terms of their melodic and harmonic intervals[4], which means that pieces do not have to be transposed into a common key or encoded with their key information in order to access their melodic or harmonic content. This approach affords a system that better relies on training data since it does not require a human analyst to label the keys of each training example. It also enables their composition system to describe modal, post-tonal and serial music, if necessary.

A focus on the key-independent and pitch-relative features such as spacing[5] and contour[6] has the advantage of historical accuracy and generalizability to more harmonically complex music as well. First, while the concept of modes and tonal centers had existed since the middle ages, functional harmony as a theory of music did not emerge until the early 19th century (Gottfried Weber was the first theorist to use roman numerals in addition to figured bass). The understanding composers like Bach had of music was in Fux's tradition of species counterpoint and the discourse surrounding Agazzari's treatises on figured bass. The twelve-tone chromatic scale did not emerge until equal temperament became common, which was during Bach's lifetime and even the idea of chords as the unit of harmony did not emerge until the work of Rameau in 1722, when Bach was 37 [23]. While arguing what Bach did or did not know about music theory would fill another paper, Bach was certainly taught composition in terms of harmonic and melodic spacing between notes.[7] Further, most western classical music is not in a single key (if it is tonal at all), so a key-relative approach to composition is not able to use pieces longer than a chorale or make use of music with ambiguous key centers.

[3] Tonality, a bit of a spectre in classical music, is the system by which notes and chords are put in hierarchy in order to create stability and instability, which introduces a system of "keys" that allow notes to have different meanings in different contexts. Bach wrote tonal music, so the same note within two different chorales may be perceived differently depending on the key. By tonality-agnostic, we mean an approach that does not require conditioning a model on key information or the expectations of tonal music.

[4] Intervals in music theory are distances between notes. Melodic intervals occur sequentially in time and refer to the contour of a single melody, while harmonic intervals occur simultaneously and refer to the spacing between notes in a chord.

[5] The spacing between two voices is the pitch distance between the notes those voices are singing, a harmonic interval.

[6] The contour of a single voice is the distances between one note that voice sings and the next, sequentially in time, a melodic interval.

[7] Here we do not mean to claim that modes and keys were not established at this point in history, since they were, but that many of the extra-musical assumptions and conventions surrounding tonal harmony and chords were not.

### 1.5 Summary

To recap, the objective of computational creativity is to create human-competitive creative systems. Bach-like chorale harmonization and composition is a well-explored domain for such research, and indeed, researchers in the past have used varied predictive models, approaches to key differences and evaluation metrics. In this paper, we propose a recurrent neural network approach to machine composition and harmonization of Bach chorales that makes use of both absolute pitch and relative pitch encodings of music, generate original melodies and chorales, harmonize existing chorale melodies, and evaluate via a survey and expert analysis.

This approach is similar to both Eck and Schmidhuber [8] and [21] in that it uses LSTM neural networks to compose music note by note but differs in that it makes use of relative-pitch encoding schemes for music. LSTMs avoid the problems of both rule-based methods, HMM methods and fixed-window neural network methods, since they require minimal expert involvement, have internal memory to establish long-term temporal dependencies and learn how long those dependencies should be. Hopefully the relative-pitch encoding schemes will allow us to avoid the problems associated with tonality as well.

## 2 Methodology

### 2.1 Note Encodings

In order to achieve our desired tonality-agnostic approach, we need to encode music in a way that uses relative pitch without focusing on key. To solve this problem, we use several different encodings of musical data to complement the information loss from encoding music in any way.

Historically, representing a piece of music as a matrix, where each column describes a (usually short) fixed time duration and each row is the one-hot encoding of a pitch that is sounded during that time duration, is common [3]; however researchers often diverge from that standard. Hild [2] represents pitches based on an encoding of the diatonic chords that contain that pitch, where each bit corresponds to a chord. Mozer [3] used a six bit one-hot encoding of a note's placement in the circle of fifths and a six-bit one-hot encoding on the chromatic circle to specify each note. Both of those systems lack sensitivity to pitch context, and in Hild's case, key-independence. Colombo et al. [20] make use of a one-hot encoding for pitch, but instead of having a fixed timestep length, they also have a one-hot encoding for duration. Hadjeres and Pachet [21] use a more modern embedding system to encode notes in a continuous vector space based on context, similar to Word2Vec [18]; however, their approach produces outputs that are entirely un-transparent and cannot be used to investigate claims about tonality-agnostic note encodings.

Since our goal is to use relative pitch models (c.f., 1.4), we make use of the following note encodings, partially illustrated in Figure 1:

1. **Voice Spacing:** A one-hot encoding of the spacing between two voices (in western music notation, this is the vertical distance between two notes sung by different sec-

tions at the same time), where each bit corresponds to a harmonic interval size and there is one encoding per timestep. The largest possible spacings occur when the reference voice is at the highest note seen in the dataset and the voice to predict is at the lowest, or vice versa. If the model outputs a spacing that would place the voice to predict above or below the highest or lowest note in the dataset, it predicts the highest or lowest note, respectively, instead.

2. **Voice Contour:** A one-hot encoding of the contour between notes in a single voice (the amount a single voice increases or decreases in pitch from one note to the next, going forward in time), where each bit corresponds to a melodic interval size. Since there is no contour before the first note in a given segment of music, there is no contour prediction on the first timestep. Similarly to the spacing encoding, the number of bits is double the size from the lowest note to highest note and outputs that would place the pitch outside of the expected pitch range is limited to the highest or lowest pitch.

3. **Beat:** A one-hot encoding of a note's position in time. Since this encoding does not represent pitch, the pitch expert that receives this encoding as input also receives a one-hot encoding of absolute pitch, starting at the lowest pitch value in the dataset and ending at the highest.

4. **Articulation:** A one-hot encoding with three values – articulate (end the previous note and start a new note), sustain (continue the previous note) and rest (end the previous note without starting a new note). This encoding is used as the input for a separate articulation model which receives the rhythm encoding and articulation encoding and predicts the articulation encoding for the next timestep.

Within each encoding system, a single voice from a chorale becomes a matrix, where each column represents a timestep and the rows represent different spacings, contours or beats.

## 2.2 Product of Experts

Like all good things in machine learning, the idea that allows us to use multiple note encodings originally comes from Geoffrey Hinton, specifically Hinton's product of experts model [6]. Instead of training a neural network to make predictions given all of the available attributes of an instance, this approach trains multiple neural networks on logically related subsets of those attributes, then multiplies their output predictions together and renormalizes. That is, instead of using a one neural network $F$,

$$Y = F(X, \theta)$$

where $X$ is the input vector, $\theta$ is a vector of parameters and $Y$ is a probability distribution over a nominal label attribute, the product of experts approach uses multiple neural networks $F_1, ..., F_n$ so that

$$Y_i = \frac{\prod_{j=1}^{n} (F_j(X_j, \theta_j))_i^{w_j}}{\sum_{c=0}^{d} \prod_{j=1}^{n} w_j (F_j(X_j, \theta_j))_c^{w_j}}$$

(where $X_j$ is a subset of the values in $X$ that are somehow related and are inputs to model $j$, $\theta_j$ is the parameter vector of model $j$ and $w$ is a vector of weights trained by backpropagation, similarly to $\theta$). As a result, the probability that a sample taken from the distribution $Y$ will be value $i$ is given by:

$$p(i|\theta_1, ...\theta_n, w) = \frac{\prod_{j=1}^{n} p_j(i|\theta_j)^{w_j}}{\sum_{c=0}^{d} \prod_{j=1}^{n} p_j(c|\theta_j)^{w_j}}$$

i.e., the probability that samples taken from each model will be value $i$ assuming the samples from all of the models are the same [6].

This approach is especially beneficial for the problem studied in this paper because the multiple neural networks used in the product of experts approach allows us to simultaneously learn over each of our different encodings of the training data and listen to models conditioned on each encoding in concert.[8]

### 2.3 Model Architecture

While we experimented with several different arrangements of product models, the final architecture is presented in Figure 2. Three spacing models, one for each other voice, along with output from a contour model and beat information model were used as input models to a product of experts. Due to the associativity of multiplication, nested products (i.e. the product of spacing models used as a single expert in the final product) don't affect the power of the model and are used solely for investigative and debugging purposes.

Each expert model consists of three LSTM layers, with 100, 200 and 100 units, respectively, followed by a layer the size of the note encoding with the softmax activation function,

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=0}^{K} e^{z_k}} \text{ for } K \text{ dimensional vector } z$$

which guarantees that each output is positive and that all outputs sum to 1. As a result, we can interpret the output as a probability distribution over possible next notes, multiply them and renormalize as a product of experts.

[8] Credit for a variant on this idea is also due to Daniel Johnson of Harvey Mudd College.

The simple generative model used as a control to evaluate the effect of the product model uses a similar architecture to each of the individual expert models. It takes an input encoding of the absolute pitch of each voice at the previous timestep (summed together, so a “four hot” encoding), concatenated with the absolute pitch of each other voice at the current timestep (summed together, a “three hot” encoding) and predicts over the current timestep of the missing voice. For consistency with the product model, the output also takes into account an articulation model, so the only difference is in the lack of a product of outputs from multiple experts trained on relative pitch encodings.

## 2.4 Training Procedure

The model is trained on a database of Bach chorales using backpropagation in minibatches consisting of 32 beat segments from each of 20 pieces from the training set. Since our goal is to compose music similar to Bach’s, the model is trained to predict each successive note in a voice. The output at each timestep is interpreted as a probability distribution, so we can calculate the likelihood that the model generates exactly the training pieces, and use that as a measure of performance. However, the likelihood is usually quite small, so we exploit the monotonicity of the log function and try to maximize the log of the likelihood instead. By convention in machine learning, though, models are discussed in terms of minimizing error rather than maximizing accuracy, so we use the negative of that log likelihood as our error. Based on a similar approach by Colombo et al. [20], we formally define the error of a parameter vector $\theta$ for neural network $F$ on a minibatch with pitch data $P$ and articulation data $A$, each a $L \times N \times M$ tensor ($L$ pieces with $N$ timesteps and $M$ possible note values/articulation values), as:

$$Error(\theta) = \sum_{i=1}^{L} \sum_{t=1}^{N} \log \Big( p(F(P_{i,t}, \theta) = P_{i,t+1} | F(P_{i,\tau}, \theta) = P_{i,\tau+1} \forall \tau < t \text{ and } A_{i,t} = a) \Big)$$

where $a$ is a constant that encodes “articulate” as opposed to sustain or rest. This training procedure is encoding-independent (assuming $F$ accounts for encoding) and can be used both to train individual expert models or a product model (assuming $\theta$ includes all of the expert parameters and product weights within a product model).

The articulation model, which is trained in the same way, though separately from the pitch models and on all timesteps regardless of articulation, predicts over the articulation encoding. The pitch models are only ever consulted about a timestep if the articulation model outputs “articulate”, which means that only timesteps with articulations in the training data are used to train the pitch model.

## 2.5 Generation Procedure

Using models trained as described above, new compositions of music can be generated as follows. Instead of feeding the timesteps from the training data in as input to the model, one voice from a chorale can be produced by iteratively sampling from the distribution of the model's output and feeding it back in as the input for the next timestep. Since input is only needed from the pitch model for timesteps where the piece articulates, we can sample from the output of the articulation model first, then use the previous timestep if it outputs sustain or no pitch at all if it outputs rest.

Since a chorale has four voices, a procedure to generate a chorale would require four models. Each of these models requires input from each other voice, creating a circular dependency. Hadjeres and Pachet [21] solve a similar problem with Gibbs sampling – they generate a piece at random, then iteratively select a note, use their model to predict that note and insert it. We follow a similar solution, though we sample voice by voice rather than note by note. In short, we generate four voices at random, then iteratively choose one voice and replace it with a generated voice. We arbitrarily repeat 100 times per outputted sample, though a smaller number of iterations may generate similar quality of output.

Once a piece is generated, we can create a score by importing into MuseScore or synthesize into audio by writing a MIDI file and using timidity [25] with a soundfont (we use MuseScore's default soundfont and play all chorale voices with piano).

Generating each voice individually gives this model an advantage over other models similar to the "steerability" of Hadjeres and Patchet's model – it is trivial to fix one or more voices to specific values and never generate a replacement for that voice. This means that not only can our our model compose completely original chorales, but it can also generate individual melodies, as well as harmonizations of existing melodies.

# 3 Implementation

Our model was implemented in the python programming language using the libraries numpy/scipy [13] for n-dimensional arrays and numerical computing, matplotlib [12] for data visualization, theano [22], for optimized mathematical computation, and theano-lstm [19], for a LSTM neural network layer implementation. Theano also includes automatic symbolic differentiation, which allowed theano-lstm to come prepackaged with several optimized gradient descent algorithms. We made use of the AdaDelta optimizer [17], in particular, to train our neural network models.

Each expert model in our software design was represented by a subclass of a GenerativeLSTM class, and a product model could have any number of GenerativeLSTMs in its product, plus an articulation model. Our main training loop first divides the dataset into training and validation sets, then iteratively chooses a minibatch of samples at random from the training set, does a forward pass through each timestep of each sample, then adjusts model parameters using gradient descent. Every hundred training iterations, we evaluate the quality of the model using a validation pass, which consists of a forward pass over the entire validation set (instead of only using a minibatch as during training) and calculate error, but do not update any parameters. If this validation error is greater than the previous validation error, training stops, since increases in error are often indicative of overfitting, when additional training will no longer improve the model.

To generate a new voice, at each timestep, each submodel is fired, the output distributions are multiplied, renormalized and sampled from. If the articulation model outputs "sustain" or “rest”, either the note encoding from the previous timestep or a zero vector, respectively, is outputted. Otherwise, the sample from the pitch models is outputted and fed back in as the input for the next timestep of the model. This process is repeated for a number of timesteps equal to the length of a random piece from the validation set.

### 3.1 Training Loss

The product model trains about 50% slower than the simple generative model, requiring around 900 training iterations to the simple model’s 600, but results in about half of the final training loss, which is to be expected from a deeper neural network model. Despite that, the simple generative model ends up with a little under 25% less final validation loss than the product model, which may indicate a more general fit to the dataset, but we do not claim that that difference is meaningful (see Figure 3).

### 3.2 Generated Probabilities

The output probabilities from each model (Figure 4) reveal both what the model was thinking before it decided on the note it did and also how certain it was each timestep. It is important to note when looking at these that they show the output of the pitch model independent of the articulation model, so the sample for any one timestep indicates the model’s answer to the question “what would the next note be here if a new note started?” Also, these models are not generating pieces of the same length (which is selected as a constant at the start of generation), so the X-axes do not correlate across the two plots.

The simple model chooses a couple pitches it likes (40, 42, 43, 45, etc.), and often is just adjusting their probabilities relative to each other. These pitches are probably the most common in the dataset, since Bach wrote in certain keys that share a lot of notes in common (C major, G major, F major) more than other, more remote keys, primarily as an artifact of the tuning systems for the harpsichord or organ he was composing on. The product model, by contrast, has a much more focused range of pitches it prioritizes, but pays much less attention to key, since it takes contour into account and small intervals are more common than large intervals. Both models notably are fairly certain ($> 90\%$ probability) of a few notes’ correctness at a few timesteps, which is notable if only relative to other models explored during the early stages of this research.

## 4 Experimental Design

The goal of this project is to create a human-competitive creative system for composing chorales. In order to measure human-competitiveness experimentally, we need a more concrete definition, both of human-competitiveness and creativity. These definitions lead us to two evaluation instruments: (1) a survey to measure human-competitiveness, and (2) an expert’s analysis to measure creativity.

## 4.1 Survey Design

First, we adopt a rather simplistic definition for human-competitiveness: music that is both (i) not easily distinguishable to the ear from that composed by a human and (ii) aesthetically appealing or enjoyable for a similar fraction of people to human composed music. A survey administered online was used to measure, in broad strokes, these two features.

The first survey question, which intended to evaluate part (i) of our definition, was designed to be an adaptation of the "Discrimination Test" evaluation metric used in Hadjeres and Pachet [21]. We asked participants "Do you think this music was composed by J. S. Bach or a computer?"

Since this paper is an undergraduate honors thesis and is not funded by a large corporation, we did not have access to as large a sample size and had to maximize the amount of information we could get out of each participant. Between the two tests used in Hadjeres and Pachet, we prioritized the discrimination test (which asked participants whether a sample was composed by Bach or a computer) over their "Perception Test" (which asked participants which of two harmonizations of the same chorale was more likely composed by Bach) since it measured human-comparability absolutely rather than relatively, so we could still compare our results for different types of outputs (for instance, solo melodies and four voice harmonizations) that would have been difficult for participants to compare directly within the test.

We added a second question, "Did you enjoy this music?" with five possible responses ranging from "I disliked this music a lot" to "I enjoyed this music a lot" to measure human comparability according to the second component of our definition. We also included a third, open-ended question, asking for feedback or explanations, to gain additional insights into why survey respondents answered the other two questions as they did.

To evaluate these components of human-competitiveness for both models, we included eight different kinds of samples to evaluate: solo melodies, chorale harmonizations and full chorale compositions by the product model discussed above; solo melodies, chorale harmonizations and full chorale compositions by a simple LSTM model (discussed below); and solo melodies and chorales from the dataset of Bach chorales. Three samples, chosen at random from samples generated by that method, were used as representatives for each category. Each participant was then given a single sample from the representatives of four of the eight categories. The full text of the survey is available in Appendix A.

## 4.2 Expert Analysis

Second, we adopt the definition of creativity put forward by Jordanous [16] (see figure 5) and in accordance with her recommendations, identify four of her fourteen components of creativity as relevant for measuring our system: 3. intention and emotional involvement, 8. originality, 9. value and 12. domain competence. The other ten components are either trivially satisfied by any similar system (2, 5, 7 and 11), too nebulously defined to be used as evaluation metrics (4 and 13) or not applicable due to the opaqueness of neural networks (1, 6, 10 and 14). A review and analysis of several samples of music by an independent expert was used to determine the extent to which we met this definition.

Using that definition, we ask a neutral expert evaluator (Dr. Rebecca Leydon of the Oberlin music theory department) to judge four example passages: a simple model harmonization, product model harmonization, simple model generation from scratch and product model generation from scratch. Examples were selected from the samples used in the survey and were converted into score form using MuseScore [24]. The full text of the email request is available in Appendix D.

# 5 Results

As discussed above, we used two instruments, a survey and expert analysis, to collect data to use to evaluate our system.

## 5.1 Survey

The first formal evaluation metric was the survey. Results for the two survey questions across different samples are presented in Figure 6. For the Bach vs. Computer test, an answer was counted as a "1" if the participant answered Bach and a "0" if they answered computer. For the quality test, answers were assigned weights from 1 to 5 based on their response, where 1 is "Disliked a lot" and 5 is "Enjoyed a lot". Superscript numbers indicate statistically significant differences between pairs within the same category (for threshold $p < 0.05$, calculated using an ANOVA and Tukey's test).

Average responses to each question for each category of samples are displayed in Table 1 in Appendix C. Breakdown of the second question, "do you enjoy this music?" is displayed in Table 2.

Unsurprisingly, the Bach outperforms all of the multi-voice samples by a wide margin. We also find that participants' quality ratings correlate positively with labeling a sample as Bach. The three music education categories did not differ much in their opinions of quality, but those who had studied Bach tended to do better on guessing Bach on all questions except for the Bach melody-only sample, where the non-musicians actually answered more accurately.

The simple model's melody output was notably not different by a statistically significant amount from the melody-only Bach sample, which more than half of participants incorrectly labeled as computer-composed. The only model outputs which were different from each other by a statistically significant amount were the samples generated from scratch, where the simple model outperformed the product model.

Many users also left comments to explain their decisions. Some examples:

- on Q1, the simple model's melody output, "Sounded like something a child would create on the piano. Kind of boring and repetitious notes."
- on Q3, the Bach melody, "Because it feels relatively repetitive, I suspect that it was composed by a computer, but I felt it had more human feeling to it than the previous sample."

- on Q5, the product model's harmonization output, "I don't have a solid reason for voting Bach, but it was perhaps the most realistic-sounding piece yet. Even though there were still some clashing notes, they still have an underlying harmonious balance and do eventually resolve."
- on Q8, the product model's full chorale generation output, "This one has a lot of notes that sound discordant to me, which makes me think it should be human because a computer should know which notes not to put together, full stop, and shouldn't be able to break the rules? But at the same time, it sounds so bad I have difficulty believing a human made it."

See Appendix B for a more representative collection.

## 5.2 Expert Analysis

The second evaluation metric was expert analysis. Our examiner was given recordings and scores for four samples, one each from the simple model and product model respectively for harmonization and generation from scratch. Select excerpts of her analysis are included below; the full text is in Appendix E.

She began her commentary with a preamble establishing criteria for domain competence:

> I take a chorale to be a musical work in (typically) four voices, with SATB ranges, with (usually) a principal melody in the highest voice.soprano lines tend to exhibit the most "florid" design and widest [range]; bass lines feature skips and leaps, especially of a 4th and 5th at cadences; alto and tenor lines are more axial in shape, with few large leaps, and they resemble each other in shape and intervallic structure more closely...

She goes on to discuss structure, that a chorale has phrases corresponding to lines of text with obvious cadences, texture, that voice parts usually change notes together, but in the middle of phrases can become differentiated, and harmony:

> The harmonic vocabulary of a chorale depends, of course, on historical and stylistic constraints (Bach's chorales are tonal, Johann Walter's are modal, Stravinsky's are serial), but regardless of style, chorales generally establish some sense of consonance and dissonance – through interval content, range, metric stability, or what have you – such that we have a sense of phrases moving through moments of instability and stability, and some criteria for "openness" and closure. This, I think, is directly related to your question about "intentionality, investment or emotional expression": the chorale will project those qualities if the phrases seem deliberately sculpted by dissonance and resolution.

Addressing individual samples, our evaluator points out numerous odd intervals and rhythms and several errors in voice leading. She finds it odd that many of the samples have articulations on every downbeat, without any longer notes to denote phrases, which contributes to a lack of any sort of harmonic syntax or structured shifts between consonance and dissonance. All of the samples have coherent individual voices and often contain lovely counterpoint. The two product model samples she finds to have better differentiated voice parts, but suffer other problems that make it difficult for her to ascribe intentionality. She does not address the originality or value of samples.

## 6 Discussion

It is safe to say that the introduction of tonality-agnostic note encoding systems did not improve human-comparability or creativity when compared with a model trained on the raw pitch data, but to call the experiments here a failure would ignore much of the nuance of our results.

First, there was no significant difference between the two models for melody generation or harmonization, but there was for full chorale generation: the simple model outperformed the product model. This may have been because the product model became too reliant on the spacing with another voice. In Bach's music, all of the voices imply each other to some extent, and assuming that one voice perfectly implies another is dangerous when dealing with generated voices. When composing a melody, this assumption held true for all three other voices, and when harmonizing, it held true for spacing with only the soprano, since it was a Bach melody, but it did not hold when generating from scratch. The simple model, by comparison, was not given spacing data explicitly and may not have developed this assumption. The output probabilities support this explanation: the simple model would always output high probabilities for a couple common notes, and would just adjust those probabilities based on context, so the reliability of other voices would not have had as much of an effect.

That said, we cannot conclude with any certainty that the product model is a flawed approach. Since it is a larger neural network with more parameters, it is more likely to train into a local minimum and may perform substantially better with more training data. Additional training with a larger chorale dataset may result in higher quality musical output.

Second, generated melody samples were much more difficult for participants to differentiate from real Bach than four part harmony samples. This may be because participants judged samples based on their dissonance levels, and unaccompanied melodies cannot be dissonant, or it may be because the generation scheme for unaccompanied melodies allowed the model to see what the other voices composed by Bach were, giving it the reliability of the assumption discussed above.

Third, the articulation model, which decided whether to articulate, sustain or rest on each timestep, created mostly believable rhythmic structures and textures in music samples. That was to be expected, since chorale rhythms are usually straightforward and easily predictable. As our expert's analysis reveals, though, that believability came at the cost of originality, since the resulting rhythmic patterns tended to be repetitive and monotonous, articulating almost every downbeat, which is a safe bet to make when predicting Bach's rhythms, but less so when creating interesting textures (c.f. Appendix E).

Fourth, relative note encodings achieved our stated goal, almost by definition, since they did not take tonality into account and allowed our model to reach similar levels of training prediction accuracy without involving tonality or music theory that did not exist during Bach's lifetime. That said, avoiding tonality did not create more aesthetically pleasing, Bach-like or creative music. This method may prove more useful, however, applied to a dataset that either contains atonal pieces or pieces with multiple key centers.

Fifth, the nature of the training data inadvertently allowed the simple model to pick up aspects of tonality, since most of the pieces were in keys that had several notes in common, that the model learned to prioritize. Importantly, though, this did not restrict the model to a specific set of pitches or impose unwanted assumptions, but it did reduce the amount of dissonance in the music it composed, especially when generating from scratch.

Sixth, synthesis, or the process of creating sound electronically from scratch, played a greater role in both participants' and our expert's evaluation of samples than expected or intended. People had a tendency, as displayed by the comments on samples, to evaluate based on timbre (instrument sound qualities) and expressiveness rather than pitches and rhythms, and were convinced that even the Bach samples were computer-composed just because they were computer-synthesized. It is possible that many participants may have rated all of the samples higher if they had been performed by humans or more sophisticated synthesizer systems rather than just MIDI.

As a suggestion for future research, another experiment with performances of generated scores or a more sophisticated synthesis technique than MIDI may lead to more focused survey results. Researchers who decide to involve human performers, though, must ensure that performances are not influenced by the performer's desire for a certain survey result. Also, since most humans would not pass a test of human-comparability where the standard is Bach, a extraordinarily good composer, evaluating computer-composed samples on a graded scale against several compositions by humans of different skill levels and musical backgrounds may prove more informative than our evaluation methods.

Finally, musical form and structure, even in a piece as short as a chorale, remains an unsolved problem. The lack of correct melodic syntax, which for a chorale means having a coherent beginning, middle and end to each phrase, was a sticking point both with survey participants and our expert for many samples, and is clearly not something LSTM networks can learn without external guidance. Future researchers may want to look into quantitative measures of consonance and dissonance and potentially train a consonance expert which is only trained on intervals' consonance values. Since dissonance is a subjective and often culture-specific phenomenon, though, more research is necessary before a model like this could be proposed.

## 7 Conclusion

We set out to create a LSTM neural network system in the line of computational creativity researchers trying to generate music based on the chorales of J.S. Bach that uses tonality-agnostic relative pitch encodings of music. We implemented such a system using Hinton's product of experts model that can compose original melodies, original harmonizations of existing chorale melodies and original full chorales after training on a corpus of those composed by Bach. We evaluate that system using a survey and an expert evaluator and found that the use of tonality-agnostic note encodings and a product model did not significantly affect the human-comparability, aesthetic quality or creative qualities of the generated melodies or harmonizations, but did negatively affect chorales generated from scratch. While we do not put forward a new state-of-the-art method to generate chorale harmonizations, we hope this research advances understanding of tonality-agnostic representations of music for computational methods and serves as an example of good evaluation methodology in the literature.

## References

[1] D. Cope. “Experiments in Musical Intelligence”. In: *Proceedings of the International Computer Music Association.* University of Michigan, 1987, pp. 174–181.

[2] H. Hild, J. Feulner, and D. Menzel. “HARMONET: a Neural Net for Harmonizing Chorales in the Style of J.S. Bach”. In: *Proceedings of the Conference on Neural Information Procesing Systems.* 1992, pp. 267–274.

[3] M. C. Mozer. “Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing”. In: *Connection Science* 6.2 (1994), pp. 247–280.

[4] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780.

[5] R. R. Spangler. “Rule-Based Analysis and Generation of Music”. PhD thesis. California Institute of Technology, 1999.

[6] G. Hinton. “Training Products of Experts by Minimizing Contrastive Divergence”. In: *Neural Computation* 14 (2000), p. 2002.

[7] M. Allan. “Harmonizing Chorales in the Style of Johann Sebastian Bach”. MA thesis. University of Edinburgh, 2002.

[8] D. Eck and J. Schmidhuber. *A First Look at Music Composition using LSTM Recurrent Neural Networks.* Tech. rep. IDSIA-07-02. www.idsia.ch/techrep.html: IDSIA, Nov. 2002.

[9] S. Phon-Amnuaisuk. “Control Languages for Harmonization Process”. In: *Proceedings of the International Conference on Music and Artificial Intelligence.* 2002, pp. 155–167.

[10] K. M. Biyikoglu. “A Markov Model for Chorale Harmonization”. In: *Proceedings of the Triennial ESCOM Conference.* 2003.

[11] M. Allan and C. Williams. “Harmonising Chorales by Probabilistic Inference”. In: *Advances in Neural Information Processing Systems 17.* Ed. by L. K. Saul, Y. Weiss, and L. Bottou. MIT Press, 2005, pp. 25–32.

[12] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: `10.1109/MCSE.2007.55`.

[13] T. E. Oliphant. “Python for Scientific Computing”. In: *Computing in Science and Engg.* 9.3 (May 2007), pp. 10–20. ISSN: 1521-9615. DOI: `10.1109/MCSE.2007.58`. URL: `http://dx.doi.org/10.1109/MCSE.2007.58`.

[14] M. A. Boden. “Computer Models of Creativity”. In: *AI Magazine* (2009), pp. 23–34.

[15] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription". In: *Proceedings of the 29th International Conference on Machine Learning*. 2012.

[16] A. Jordanous. "A Standardized Procedure for Evaluating Creative Systems". In: *Cognitive Computation* 4 (2012), pp. 246–279.

[17] M. D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: *CoRR* abs/1212.5701 (2012). URL: `http://arxiv.org/abs/1212.5701`.

[18] T. Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781 (2013). URL: `http://arxiv.org/abs/1301.3781`.

[19] J. Raiman. *Small Theano LSTM Recurrent Network Module*. Dec. 2014. URL: `https://github.com/JonathanRaiman/theano_lstm`.

[20] F. Colombo et al. "Algorithmic Composition of Melodies with Deep Recurrent Neural Networks". In: *ArXiv e-prints* (June 2016). arXiv: `1606.07251 [stat.ML]`.

[21] G. Hadjeres and F. Pachet. "DeepBach: a Steerable Model for Bach chorales generation". In: *CoRR* abs/1612.01010 (2016). URL: `http://arxiv.org/abs/1612.01010`.

[22] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: `http://arxiv.org/abs/1605.02688`.

[23] I.D. Bend and A. Pople. "Analysis". In: *Grove Music Online*. Oxford, UK: Oxford University Press, 2017.

[24] T. Bonte, N. Froment, and W. Schweer. *MuseScore: Music Notation and Composition Software*. URL: `https://github.com/musescore/MuseScore`.

[25] T. Toivonen and M. Izumo. *TiMidity++ Software Midi Synthesizer*. URL: `http://timidity.sourceforge.net/`.

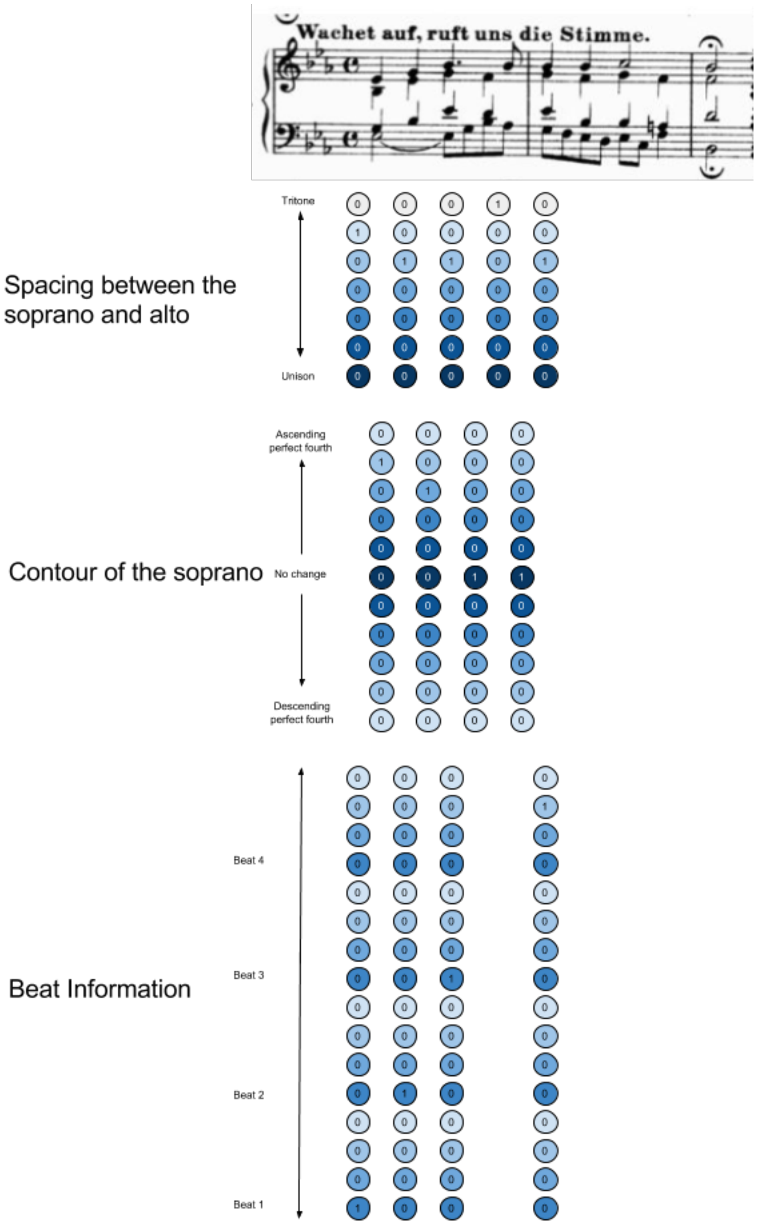

Figure 1: Example of pitch and rhythm encoding schemes used. Note that for simplicity of the diagram, some timesteps are omitted, as there are actually four timesteps per quarter note of music.

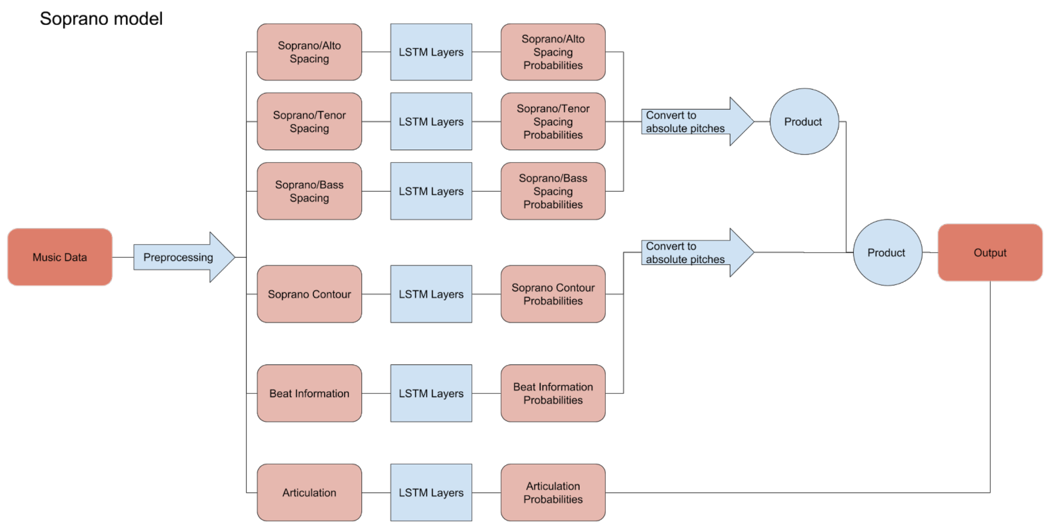
Soprano model
Music Data
Preprocessing
Soprano/Alto Spacing
Soprano/Tenor Spacing
Soprano/Bass Spacing
Soprano Contour
Beat Information
Articulation
LSTM Layers
LSTM Layers
LSTM Layers
LSTM Layers
LSTM Layers
LSTM Layers
Soprano/Alto Spacing Probabilities
Soprano/Tenor Spacing Probabilities
Soprano/Bass Spacing Probabilities
Soprano Contour Probabilities
Beat Information Probabilities
Articulation Probabilities
Convert to absolute pitches
Convert to absolute pitches
Product
Product
Output

Figure 2: Data flow in the model of one voice. Similar models were used for the other three choral voices.

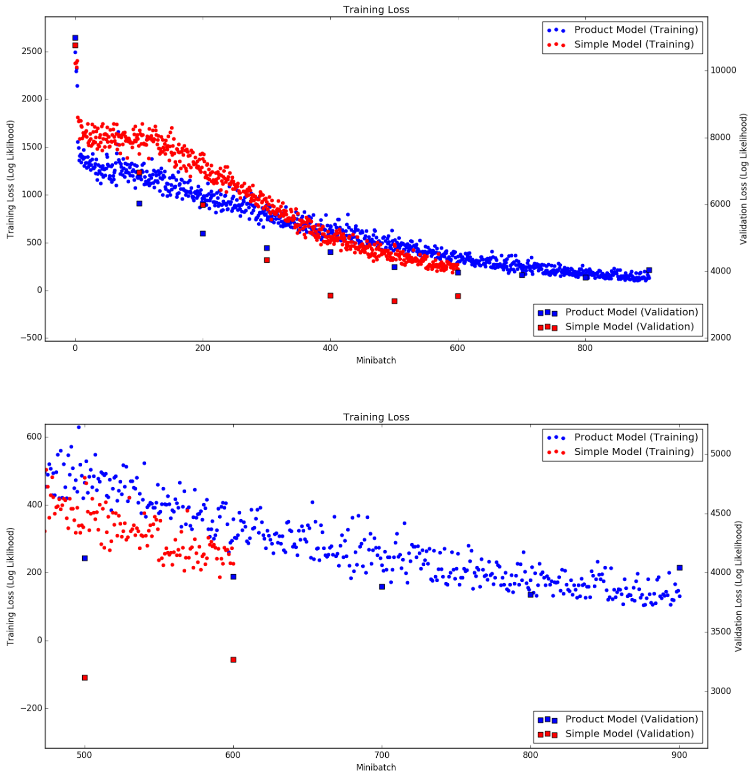

Figure 3: Training/Validation loss for both models by minibatch.

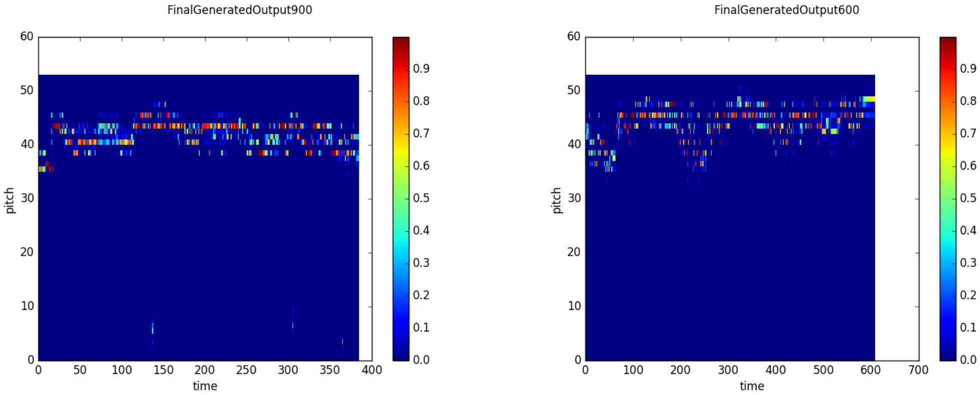

Figure 4: Probability distributions over next pitches for the product model (top) and simple model (bottom). The heatmap indicates probability.

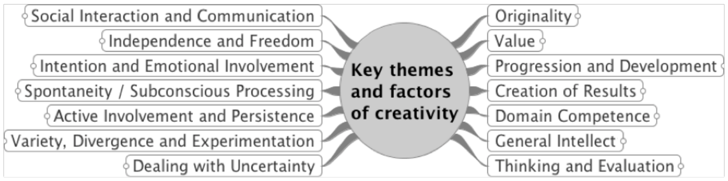
Key themes and factors of creativity
Social Interaction and Communication
Independence and Freedom
Intention and Emotional Involvement
Spontaneity / Subconscious Processing
Active Involvement and Persistence
Variety, Divergence and Experimentation
Dealing with Uncertainty
Originality
Value
Progression and Development
Creation of Results
Domain Competence
General Intellect
Thinking and Evaluation

Figure 5: A visual representation of Jordanous' definition of creativity [16].

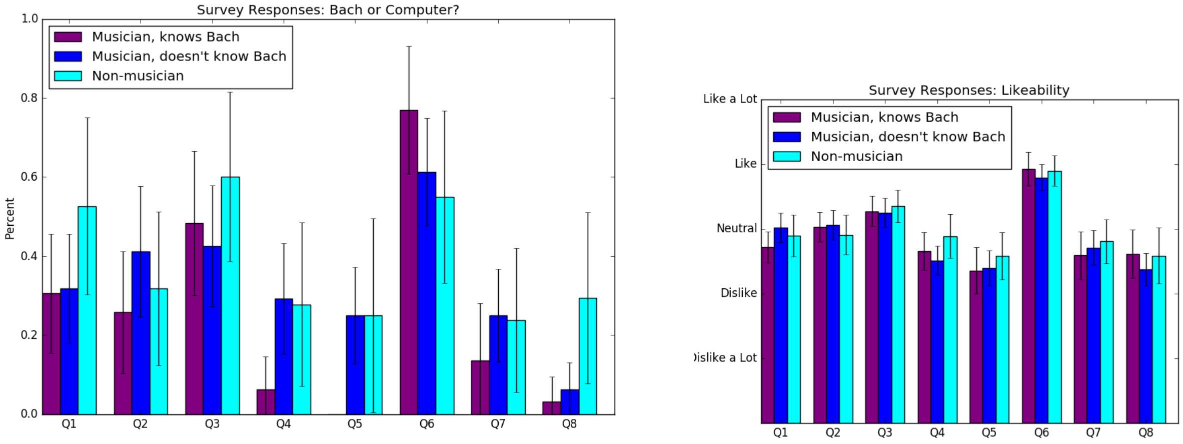

Figure 6: Survey responses visualized. Groupings correspond to questions about eight categories of samples, The three bars for each question correspond to subsets of the sample split according to their self-identification of their musical background, either "I am a musician or music enthusiast and have studied the music of J.S. Bach," "I am a musician or music enthusiast who has not studied the music of J.S. Bach" or "I am not a musician or music enthusiast," in that order.

# A Survey Text

Thank you for agreeing to help me complete my Honors project! First, please answer some demographic questions, then answer a few questions related to music samples, which will be in the form of links to Soundcloud pages. Make sure you either have headphones or are in a space where you can listen to music before beginning! If you have any questions or issues, feel free to contact me at sgoree [AT] oberlin.edu!

What is your age?

- Under 18 years old
- 18-24 years old
- 25-39 years old
- 40-54 years old
- 55-69 years old
- 70 years old or older

Which of the following best identifies you?

- "I am a musician or music enthusiast who has studied the music of J. S. Bach"
- "I am a musician or music enthusiast, though I have not studied the music of J. S. Bach"
- "I am not a musician or music enthusiast"

Please identify your race/ethnicity:

- White
- Hispanic or Latino
- Black or African American
- Native American or Alaska Native
- Asian, Native Hawaiian or Pacific Islander
- Other

Please identify your gender:

- Male
- Female
- Non-binary
- Other

<Participants are asked the following questions four times>

For the next three questions, please base your answers on this sound sample:

<Below is an embedded SoundCloud player for one of 24 possible sound samples, three from each of the eight categories discussed in the results section>

Do you think this music was composed by J. S. Bach or a computer?

- Bach
- Computer

Did you enjoy this music?

- I enjoyed this music a lot
- I enjoyed this music
- I neither enjoyed nor disliked this music
- I disliked this music
- I disliked this music a lot

Do you have any comments regarding the believability or aesthetic quality of this sample?

## B Select Survey Responses

1. simple melody
   - Sounded like something a child would create on the piano. Kind of boring and repetitious notes.
   - I voted Bach simply because the simplicity makes it seem like I'm "supposed" to guess computer.
2. product melody
   - felt more human but lacked natural dynamics based on humans hitting the keys with different forces
   - The melody made it difficult to find the tonality. I couldn't get a sense of what the root was.
3. Bach melody
   - Because it feels relatively repetitive, I suspect that it was composed by a computer, but I felt it had more human feeling to it than the previous sample.
   - Coherent phrases!! Much better. This music not only sounds better, but it makes a lot more sense than any of the others so far.
4. simple harmonization
   - Is this being played on a computer as opposed to on a piano or electronic keyboard and then recorded? It has an industrial sound that made it hard to figure out. Could be Bach. Not sure. Thought I heard a familiar tune there.
   - Very Charles Ives.

- Aesthetically, I enjoyed it because of how it flirted between dissonance and familiar consonance, but it was nowhere close to following the ”traditional” harmonic conventions of Bach so as far as pure believability goes, it didn’t sell me.

5. product harmonization

   - I don’t have a solid reason for voting Bach, but it was perhaps the most realistic-sounding piece yet. Even though there were still some clashing notes, they still have an underlying harmonious balance and do eventually resolve.

   - Ugh sounded like someone was just pounding randomly on the piano...very uneven and unpleasant

   - there was little to no overarching structure, the rhythm was disjointed and unresolved dissonance prevailed

6. Bach chorale

   - The performance (which I suspect may be computer-generated rather than played) is so unyielding rhythmically (i.e. devoid of accent) that it was annoying to listen to. (I wrote that I enjoyed it, which I did, but through a kind of filter of annoyance. I couldn’t have taken much more of it.) I base my “computer” answer partly on not recognizing it, and partly on a kind of hyper-regularity to the composition that doesn’t remind me of JSB at all.

   - The structure was strong in this one, and what I know to expect from Bach.

7. Simple generation

   - The having multiple chords made me think it was Bach at first, but now I think it’s too jumbled and discordioate.

   - Like the previous two, nothing Bach-like about the harmonic or rhythmic idiom. It has a kind of fluidity that I liked better than the previous two.

   - You could tell the composition was Bach but some notes fell into each other & some were jarring.

8. Product generation

   - Very atonal, tho some of the rhythmic aspects were similar to a Baroque style

   - This one has a lot of notes that sound discordant to me, which makes me think it should be human because a computer should know which notes not to put together, full stop, and shouldn’t be able to break the rules? But at the same time, it sounds so bad I have difficulty believing a human made it.

## C Survey Data

Table 1 displays average score for each sample category on each survey question (superscripts indicate stastically significant differences). Questions are numbered as follows:

1. Melody generated by the simple model

| Question | Melody Only | | | Harmonization | | Full Bach | Generation | |
|---|---|---|---|---|---|---|---|---|
| | Simple | Product | Bach | Simple | Product | Chorale | Simple | Product |
| Bach vs. Computer | .35 | .3258$^{1}$ | .4858$^{1}$ | .2043$^{2}$ | .1685$^{3}$ | .6458$^{2,3,4,5}$ | .2188$^{4,6}$ | .0938$^{5,6}$ |
| Quality | 2.876$^{1}$ | 3.0$^{2}$ | 3.291$^{1,2}$ | 2.629$^{3}$ | 2.412$^{4}$ | 3.859$^{3,4,5,6}$ | 2.693$^{5}$ | 2.505$^{6}$ |

Table 1: Survey results. A breakdown by musical background is visualized in figure 6.

2. Melody generated by the product model
3. Melody written by Bach
4. Harmonization via the simple model of a melody by Bach
5. Harmonization via the product model of a melody by Bach
6. Chorale written by Bach
7. Chorale generated from scratch via the simple model
8. Chorale generated from scratch via the product model

Table 2 displays a breakdown of the quality question by answer.

| Question | Melody Only | | | Harmonization | | | Generation | |
|---|---|---|---|---|---|---|---|---|
| | Simple | Product | Bach | Simple | Product | Bach | Simple | Product |
| Enjoyed a Lot | 0.0 | 0.0 | 1.12 | 0.0 | 1.12 | 13.54 | 2.08 | 1.04 |
| Enjoyed | 20.0 | 23.60 | 35.96 | 13.98 | 10.11 | 61.46 | 18.75 | 15.63 |
| Neutral | 52.0 | 52.81 | 53.93 | 40.86 | 34.83 | 21.88 | 32.29 | 29.17 |
| Disliked | 24.0 | 23.60 | 7.87 | 39.78 | 35.96 | 3.13 | 40.63 | 39.58 |
| Disliked a Lot | 4.0 | 0.0 | 1.12 | 5.38 | 17.98 | 0.0 | 6.25 | 14.58 |

Table 2: Breakdown of different answers to the quality question for each sample category (all numbers are percents).

## D Email Request

Hi Dr. Leydon,

Thank you again for agreeing to help with this part of my project! Attached are four short compositions. Recordings (synthesized from midi files) are at the links provided below.

I'd like you to answer the following questions in a sentence or two for each sample, treating them as though they were the product of a novice (human) composer, explaining your answers with examples when possible:

1. To what extent does this piece demonstrate domain competence? (i.e. is it recognizably a chorale, does it follow the conventions of chorale composition, etc.)

2. To what extent does this piece demonstrate the intentionality, investment or emotional expression of the composer?

3. To what extent does this piece demonstrate original thinking or come across new relationships between existing concepts?

4. To what extent is this piece valuable as a piece of music or work of art?

I am interested more in the ”why” than the evaluation (don’t worry about my personal feelings or the feelings of the composer). I know how these pieces sound and I want your honest opinions.

## E Full Evaluator Response

Note that:

- sample 11 is the harmonization output from the simple model
- sample 13 is the harmonization output from the product model
- sample 21 is the full chorale generation output from the simple model
- sample 24 is the full chorale generation output from the product model

PREAMBLE

Regarding the issue of “domain competence,” it seems necessary to establish the general criteria for a “chorale-like” music. I take a chorale to be a musical work in (typically) four voices, with SATB ranges, with (usually) a principal melody in the highest voice. Historically, chorales were 4-voice settings of recognizable hymn tunes, with the tune assigned to the highest voice and harmonized by the lower voices. While the chorale texture was later adapted for instrumental ensembles (e.g., brass chorales) with the voice ranges expanded, the model remains vocal music, and the distinctive melodic style of the SATB parts is retained: soprano lines tend to exhibit the most “florid” design and widest tessitura; bass lines feature skips and leaps, especially of a 4th and 5th at cadences; alto and tenor lines are more axial in shape, with few large leaps, and they resemble each other in shape and intervallic structure more closely than either voice resembles the soprano or bass part. Voice parts stay within specific ranges, and avoid immediate boundary “overlaps” (e.g., tenor voice moving to a note above where the alto just sang) and voice crossings.

Phrase lengths of a chorale correspond to lines of text, and even instrumental chorales tend to show phrase periodicity, and phrases often fall into phrase lengths of four and eight bars, with obvious cadences separating them.

Above all, the texture of a chorale is probably its most distinctive trait. Voice parts frequently move in homophonic synchrony, but not exclusively: phrase beginnings and endings feature synchronized articulations, but the voices tend to become more contrapuntally differentiated in the interior of phrases. An expert chorale composer like Bach may do something like this: if the predominant rhythm of all the voices is even quarter notes, the alto may introduce faster moving 8th notes, then the tenor, then the bass, so that the rhythmic activity circulates through the ensemble. This serves the function of maintaining voice independence, and draws a listeners attention to first one voice, then another.

The harmonic vocabulary of a chorale depends, of course, on historical and stylistic constraints (Bachs chorales are tonal, Johann Walters are modal, Stravinskys are serial), but regardless of style, chorales generally establish some sense of consonance and dissonancethrough interval content, range, metric stability, or what have yousuch that we have a sense of phrases moving though moments of instability and stability, and some criteria for “openness” and closure. This, I think, is directly related to your question about “intentionality, investment or emotional expression”: the chorale will project those qualities if the phrases seem deliberately sculpted by dissonance and resolution.

There are some of the features I would consider relevant to assessing “domain competence” in these examples.

SAMPLE 11

The soprano tune is convincing (and pleasant) as a modally inflected melody, up to the first half of measure 7. The pronounced modal quality of this tune sets up some expectations of how the harmonic style may play out in this piece.

The alto line, on its own, is plausible; the oscillating tritone (C#-G) in measure 3 is odd, but is followed by a resolution to D-F. But many of the melodic intervals seem unidiomatic: the augmented 2nd in m.4, the tritone D-G# followed by and augmented 2nd in m.5; the major7th leap in measure 6these are not “voice-like”.

The tenor has a similarly unsingable intervallic structure: measure 2 goes down an 8ve, up a minor 9th, down an aug 2nd, for example.

The bass likewise has some weird melodic intervalsin m 2 for example, although these are more acceptable since they are part of a sequence (B-C-A-Aflat, D-C#-Bflat-A). But the subsequent leap up a 6th followed by a descending 10th wrecks the pattern. Measure 4 takes the line from C up a minor 9th to C# then down a major 9th to B, which does not cohere as a single voice. The bass does start to sound more coherent after that, however, from the 3rd beat of m.4 until the end.

The overall texture is appropriately homophonic at the beginning, but the rhythmic “action” of the 16th notes might be spread through the ensemble to better differentiate the voice parts. The texture at mm5-6 is more idiomatic, for instance. I think I would enjoy this piece more if there were some relief from the simultaneous down-beat articulations in all voices all the time.

The ending sounds like an ending!

SAMPLE 13

To me the texture of this one sounds less frenetic than Sample 11, with the 16th-note action distributed through the layers, but again it is almost unrelievedly homophonic, with most of the voices articulating 8th-note pulses (except in measures 1, 4, 7, and 10)

Soprano tune begins as a plausible modal melody–although the E on the “&” of beat 2 is unexpected (I wanted an F there, as in a tonicization of III in F major). The tune kind of goes off the rails at the end of m4 into m5, with the descending m9th followed by a leap up a 10th. Eighth-note rests on the off beats are strange, but could I suppose be motivated by a text. Likewise, the 16th-dotted 8th figure in measure 6 is rhythmically peculiar.

The alto is somewhat convincing, although it moves in parallel 8ves with the soprano in the middle of bar 2 and in m. 6. Voice overlaps with the soprano are in measure 3 into 4, and end of m.5. The tritone leap in measure 7-8 is possibly outlining a harmony, but the one at the end is less acceptable.

Tenor line is fairly coherent until measure 7-8. The downward leaps of a 6th (e.g., m.7, end of m. 11) are atypical of inner voices.

The bass makes parallel octaves with the tenor at the beginning. The line overall suffers from a general "gap-fill" problem, where large leaps are expected to be followed by a change of direction and return to stepwise motion. The leaps by 7th and 9th in measures 6-8 dont help matters.

I won't address harmonic syntax here, since that eludes meespecially at the end. Why did it end when it did?

SAMPLE 21

The rhythmic structure of the soprano in measures 1 to 4 suggests a meter of 3/2 (starting with a down beat) rather than the notated 4/4, so there is a conflict between the notation and the sound. Cross relations and the unresolved tritone in measure 4 and 5 are pretty wild. The dotted 8th-16th figure followed by a rest is peculiar, but, again could be motivated by a particular text setting (same thing is in the alto and tenor in m 4, and in the alto in measure 15). The soprano tune repeatedly hits a high E: m2, 3, 5, 6, 7, 12, 13, 22, 23, and so it doesnt conform to the usual arch-like shape of a typical soprano. The melody in measures 23 through 25, however is very nicely shaped, and I especially like the suspension and ornamented resolution over the bar line into m 25.

The alto has some lovely counterpoint with the soprano in measures 1-3, although the tritone leap at the beginning of measure 4 kind of ruins things. After that the alto presents a pretty good tune through measure 13, with nice contrapuntal relationships to the soprano (except for parallel unisons in m 5 and the crunch on the 3rd beat of measure 11.) Overall the alto line fairly idiomatic.

The tenor, on the other hand, has issues: the line in m3 and following goes up a tritone, up a M2nd, down a M7th, up a 6th, down a 9th, etc. Large unintegrated leaps happen at m 11-12 and m.15.

Bass line has a few funny tritone and augmented seconds, but a tolerable melodic shape overall– until mm 14-16; also the leaps in m 23 are too large to cohere into a single "vocal percept." Repeated bass note over the bar line in measures 3-4 disrupts the metercompounding the metric confusion of the soprano at the opening.

It seems like there are fewer issues with voice crossings and overlaps in sample 21, compared with samples 11 and 13, and also a better distribution of 16th-note "action" among the ensemble. If we took out the tenor line, Id say this one shows the greatest domain competence of the three samples so far.

SAMPLE 24

Right off the bat, the rhythms are peculiar for a choraleespecially the non-synchronized opening. Again, I suppose this might be justified by a text rhythm. (Are these setting

Hungarian texts?).

The tenor goes below the bass at the end of m1 into 2. There are some other voice overlaps

In some ways, this sample seems least close to the conventions of a chorale, but it is does have fairly well differentiated voice partsmy attention is drawn by the rhythmic activity to first one voice, then another

Sections do sound “intentional” but not in a smooth “rising-and-falling-tension” kind of way across the entire piece. I feel like a different timbre might change my assessment of this one.