Bacher than Bach? On Musicologically Informed AI-based Bach Chorale Harmonization

Alexander Leemhuis, Simon Waloschek, and Aristotelis Hadjakos

Center of Music and Film Informatics
Detmold University of Music, Germany
OWL University of Applied Sciences and Arts, Germany
{a.leemhuis,s.waloschek,a.hadjakos}@cemfi.de
http://www.cemfi.de

Abstract. Writing chorales in the style of Bach has been a music theory exercise for generations of music students. As such it is not surprising that automatic Bach chorale harmonization has been a topic in music technology for decades. We suggest several improvements to current neural network solutions based on musicological insights into human choral composition practices. Evaluations with expert listeners show that the generated chorales closely resemble Bach's harmonization style.

Keywords: Bach Chorale Harmonization · Deep Learning · Beam Search

1 Introduction

Chorales by J.S. Bach traditionally play an important role in Western music education. Concise voice leading techniques and precepts such as the often quoted *prohibition of parallel fifths* make these chorales interesting as subject in music theory. But they are also interesting for computational music analysis and generation. Especially automatic harmonization of melodies, i.e., producing a four-part chorale given the soprano part, has been a topic for a long time.

In 1986, the first significant attempt was made: The CHORAL system [4] used over 270 hand-engineered rules for harmonization. Later, focus shifted from rule-based systems to neural networks [15,11]. In 2002, the usage of Recurrent Neural Networks (RNN) and Long Short-Term Memory cells (LSTM) [7] by Eck and Schmidhuber [5] specifically addressed the sequential nature of music and produced state-of-the-art results at that time. A decade later, statistical models like Hidden Markov models and Bayesian networks were developed [1,16,13]. Recent solutions such as BachBot [10] and DeepBach [6] again use LSTMs and incorporate metadata such as information on fermatas or metrical positions of notes to enhance the results.

Although various music theory concepts have been applied for evaluation of the resulting chorales, the actual human composition process has not yet been used for modeling neural networks. We therefore propose a Convolutional Neural Network (CNN) architecture that follows—to some extent—workflows that are documented and commonly recommended in music theory literature and taught in music theory classes for writing four-part chorales.

Expert listening tests with musicologists and music majors indicate that some of our generated harmonizations are more Bach-like than the originals, in the sense that they were believed to be the work of Bach even in direct comparison to the master's original harmonization of the same soprano part.

2 Musicologically Informed Harmonization

Contemporaries of the Baroque epoch as well as modern experts recommend to start four-part harmonization by elaborating a bass part given the soprano part, see for example [3,8]. The bass part is not only considered one of four equitable voices but also an indicator of the tonal skeleton: Once the bass line is determined, the structure of the chorale is mostly set. Only small leeway is left for the middle voices that are formed in a second step and can be very plain, solely blending into the harmonic progression [3, p. 255]. G.Ph. Telemann emphasizes in [14] that the alto part should be written before the tenor part so that the closest possible voicing can be accomplished. The advantages of generating the bass line first in generative systems have already been discussed [16].

Particular attention should be paid to the ends of musical phrases, typically marked by fermatas. Such phrases oftentimes end with rather canonical cadences and thus should be prepared in advance as Daniel suggests [3, p. 159]. Daniel also argues, that in many cases there is only one solution for a valid choice of alto and tenor notes [3, p. 256]. Therefore, sometimes during harmonization the choices of specific notes lead to dead ends in a sense that further voice development breaks common voice leading rules. These problems are commonly solved by simply going back and revising certain notes.

In summary, expert knowledge teaches us to use the following strategies when harmonizing Bach chorales:

- Generate the bass part first given the soprano part
- Support close voicings by choosing tenor notes after the alto
- Give enough context to allow for correct cadences
- Allow changes to previously generated notes

The following sections describe how these insights were integrated in our approach.

2.1 Data Processing & Augmentation

Symbolic score data is retrieved from and processed with the music21 [2] framework for Python. Besides offering various possibilities to process symbol music, it also includes a corpus with numerous chorales composed by Bach. To augment this dataset, all pieces are transposed up and down to different keys. Transpositions are limited in such way that no voice part exceeds the tonal range as used by Bach in order to ensure generation of "singable" results.

The smallest time unit used in Bach chorales is a semiquaver. Therefore we use a semiquaver time resolution to retain all information. For each time step, we compute one-hot vectors per part. The individual vectors can encode one of three slightly different events for each time step:

– **New note** If a new note starts at the given time step, its pitch is encoded.
– **Rest** Rests are handled as if they were notes with a special pitch value.
– **Continuation** In case that a note or rest is tied, i.e., not finished yet, we set a special continuation flag.

Additional score information (hereinafter called metadata) such as the current time position within a measure, the overall key of the choral, its time signature and the position of fermatas are also fed as one-hot vectors into the network.

We are aware, that Bach sometimes used the same melody to compose several different chorales. Therefore, it may happen that a specific melody has been present in the training as well as the test dataset due to random splitting of the dataset. Since the harmonizations in such cases are still different, we follow the practice of similar generative systems [6,10] and do not take this circumstance further into account.

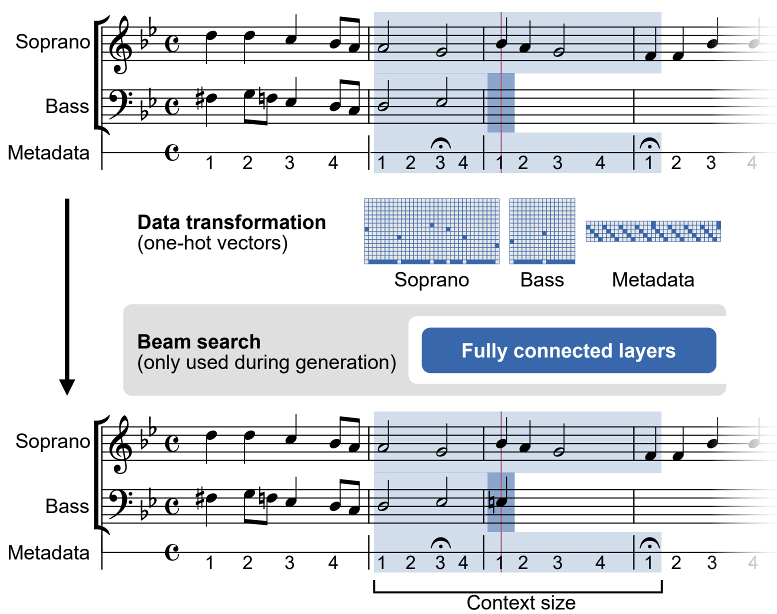

Fig. 1. Scheme of the bass part generation. The one-hot encoded data is fed into several fully connected layers to generate the output for a single time step. Afterwards, the context window is shifted by one step into the future. (Context size shown in blue is deliberately reduced compared to the actual implementation to enhance readability.)

2.2 Network Architecture

Our proposed architecture features three similar consecutive networks. The first network creates a single bass note. It takes a frame of the soprano part, metadata and the prior bass notes as an input. After the entire bass line is generated, two networks are alternated to generate alto and tenor notes based on the soprano/bass part, metadata and the previously generated middle voices. Each part is generated using only a single hidden layer of size 650. Input and output layer dimensions are defined by the individual pitch range of each part. The output layers use softmax nonlinearities, all other layers use SELUs [9]. The ordering of note generation is as follows:

1. The entire bass line is generated first. A bass event b_i depends on the soprano and metadata in a local context of ± 32 time steps $s_{i-32:i+32}, m_{i-32:i+32}$ and 32 previous bass events $b_{i-32:i-1}$ (see Fig. 2.1). We use 32 steps (8 quarter notes) as a context as it provides a sufficient look ahead to prepare cadences as suggested by Daniel [3, p. 159]. The probability model for predicting b_i is thus
$$p(b_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i-1}).$$
2. After the bass line and thus the harmonic outline is completed, tenor and alto voice are generated from time step to time step. The alto prediction a_i is generated based on soprano and metadata context as above but with current and future bass events, which have been generated in the previous step, as well as previous alto events $a_{i-32:i-1}$ and tenor events $t_{i-32:i-1}$. The underlying probability model is thus
$$p(a_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i+32}, a_{i-32:i-1}, t_{i-32:i-1}).$$
3. The tenor is generated similar to the alto, but it also depends on the alto note generated in the current time step i, i.e., it depends on $a_{i-32:i}$:
$$p(t_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i+32}, a_{i-32:i}, t_{i-32:i-1}).$$

2.3 Beam Search

For every time step i, our network predicts the probabilities $p(b_i|\cdot)$, $p(a_i|\cdot)$ and $p(t_i|\cdot)$ conditioned on the local context. We want to find the sequence that maximizes the total probability, which is the product of the probabilities for each choice[1]
$$\prod_{i=0}^{N} p(b_i|\cdot) \; \prod_{i=0}^{N} p(a_i|\cdot)\; p(t_i|\cdot).$$

A greedy approach would select pitches with maximal probability at every prediction step. However, since future predictions depend on previous ones

[1] The multiplication is split in two parts to emphasize that the entire bass line is created first.

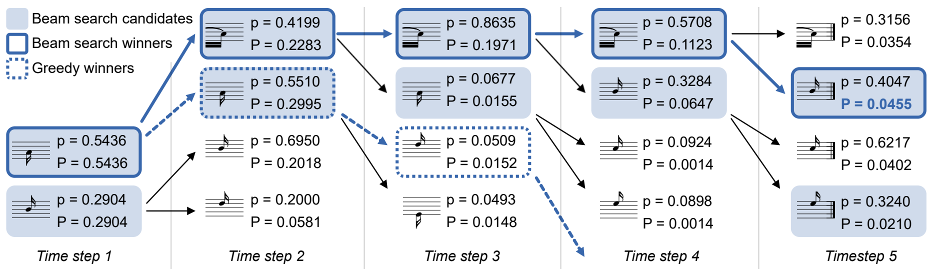

Fig. 2. Example of beam search for bass part with beam width of 2 in comparison to a greedy approach. P denotes the total probability of the branch, p denotes the conditional local probability.

(see Sec. 2.2), always choosing the highest local probability option can lead to suboptimal total probability of the sequence.

We therefore use *beam search*[12] to find solutions that help maximizing the total probability of the sequence. Beam search is a best-first search algorithm where only a fixed number of candidate alternatives are maintained to limit runtime and memory requirements. Previous work in Bach chorale harmonization has in fact suggested to use beam search, see [10]. Up to now this has, however, not been implemented and evaluated. Fig. 2 provides a graphical example of how we employ beam search for generation of the bass part. The alto and tenor parts are generated in a similar manner.

3 Generation Results

At first glance, the chorales produced exhibit similarities to original Bach chorales. Two of the generated chorales were randomly chosen and given to Lydia Steiger, music theory teacher at the Detmold University of Music, for an in-depth musical analysis. She provided the following feedback:

– In several places voice leading rules were violated.
– The algorithm lacked sensitivity for musical tension and therefore sometimes choses a plain solution in places were a more sophisticated composition would have been more appropriate.
– The network uses common musical phrases used by J.S. Bach. In some places, the algorithm split these phrases arbitrarily across voices.

Further development of this approach should aim to address these shortcomings. All generated chorales of the test dataset and other pieces can be reviewed online at the project homepage[2], see Fig. 3 for a generation example.

[2] See `http://www.cemfi.de/research/bachnet` (accessed: 2019-09-03)

Fig. 3. Generated harmonization given the melody from "Ich ruf zu dir, Herr Jesu Christ" (BWV 177.5) by Bach.

4 Evaluation

We also evaluated our network with two online listening tests. The first test *without beam search* was conducted with the help of music majors. Thus we expect a high degree of familiarity with Bach chorale harmonization. The test presented paired samples consisting of (A) the original four-part chorale by Bach and (B) our generated harmonization using the same soprano part. Participants first had to give a self-assessment about their familiarity with Bach chorale harmonization and were then asked to identify the original Bach chorale for each pair. In case participants were unsure, question could be skipped. In 61% of the presented pairs, the participants could correctly identify the Bach work. 39% misjudged our generated pieces to be composed by Bach or skipped questions (see Fig. 4). Interestingly, 5 of 17 generated chorales could not be correctly identified by the majority of the participants.

After implementing beam search, we once more evaluated our solution. Since the results had subjectively improved, we decided to evaluate our network with participants that had an even greater expertise by directly addressing professional musicologists. Apart from the new harmonizations, the same online survey was used. Only 66% could distinguish the Bach pieces from the artificial ones, 34% chose the generated harmonizations or gave no answer. Although some of the musicology experts might be familiar with the *exact* Bach chorale, still 3 of

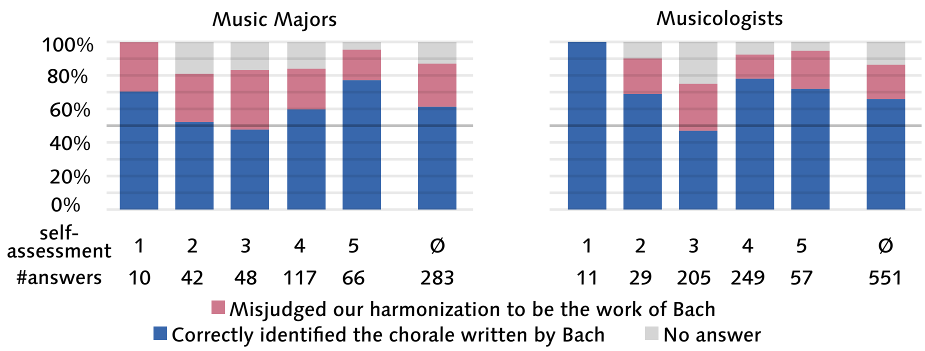

Fig. 4. Summarized results of the online evaluation. The chart shows how both participating groups scored in identifying the original Bach chorale given a generated harmonization as well as the master's work broken down by the self-assessment given. 5 corresponds to a high familiarity with Bach chorale harmonization, 1 corresponds to a low familiarity. 68 music majors and 127 musicologists participated.

17 generated chorales were not correctly identified by more than 50% of the participants. One chorale was even preferred over the authentic work.

To conclude this paper, we encourage future research on harmonization and automatic composition based on neural networks to take the human music creation process into account. The question, why several pieces sound more Bach-like than the original works even to experts could also be an interesting topic: What is it that deceives the listener and makes these chorales sound "bacher than Bach"?

References

1. Allan, M., Williams, C.: Harmonising Chorales by Probabilistic Inference. Advances in Neural Information Processing Systems **17** (2005)
2. Cuthbert, M.S., Ariza, C.: music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data (2010)
3. Daniel, T.: Der Choralsatz bei Bach und seinen Zeitgenossen: Eine historische Satzlehre. Verlag Dohr, Köln-Rheinkassel, Germany (2000)
4. Ebcioglu, K.: An Expert System for Chorale Harmonization. In: Proceedings of the 5th AAAI National Conference on Artificial Intelligence (1986)
5. Eck, D., Schmidhuber, J.: A First Look at Music Composition using LSTM Recurrent Neural Networks. Instituto Dalle Molle di studi sull' intelligenza artificiale (2002)
6. Hadjeres, G., Pachet, F., Nielsen, F.: Deepbach: A Steerable Model for Bach Chorales Generation. In: Proceedings of the 34th International Conference on Machine Learning (2017)
7. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9** (1997)
8. Kaiser, U.: Der vierstimmige Satz: Kantionalsatz und Choralsatz. Bärenreiter, Kassel, Germany, 4th edn. (2015)

9. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-Normalizing Neural Networks. Advances in Neural Information Processing Systems **30** (2017)
10. Liang, F.T., Gotham, M., Johnson, M., Shotton, J.: Automatic Stylistic Composition of Bach Chorales with Deep LSTM. In: Proceedings of the 18th International Society for Music Information Retrieval Conference. Suzhou, China (2017)
11. Mozer, M.C., Soukup, T.: Connectionist Music Composition Based on Melodic and Stylistic Constraints. Advances in Neural Information Processing Systems **3** (1991)
12. Norvig, P.: Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. Elsevier Science (1992)
13. Suzuki, S., Kitahara, T.: Four-part Harmonization Using Bayesian Networks: Pros and Cons of Introducing Chord Nodes. Journal of New Music Research **43**(3) (2014)
14. Telemann, G.P.: Fast allgemeines Evangelisch-Musicalisches Lieder-Buch. Philip Ludwig Stromer, Hamburg, Germany (1730)
15. Todd, P.M.: A Connectionist Approach to Algorithmic Composition. Computer Music Journal **13**(4) (1989)
16. Whorley, R.P., Rhodes, C., Wiggins, G., Pearce, M.T.: Harmonising Melodies: Why Do We Add the Bass Line First? (2013)