SCORE-INFORMED SOURCE SEPARATION OF CHORAL MUSIC

Matan Gover
Schulich School of Music, McGill University
matan.gover@mail.mcgill.ca

Philippe Depalle
Schulich School of Music, McGill University
philippe.depalle@mcgill.ca

ABSTRACT

Choral music recordings are a particularly challenging target for source separation due to the choral blend and the inherent acoustical complexity of the 'choral timbre'. Due to the scarcity of publicly available multi-track choir recordings, we create a dataset of synthesized Bach chorales. We apply data augmentation to alter the chorales so that they more faithfully represent music from a broader range of choral genres. For separation we employ Wave-U-Net, a time-domain convolutional neural network (CNN) originally proposed for vocals and accompaniment separation. We show that Wave-U-Net outperforms a baseline implemented using score-informed NMF (non-negative matrix factorization). We introduce *score-informed Wave-U-Net* to incorporate the musical score into the separation process. We experiment with different score conditioning methods and show that conditioning on the score leads to improved separation results. We propose a 'score-guided' model variant in which separation is guided by the score alone, bypassing the need to specify the identity of the extracted source. Finally, we evaluate our models (trained on synthetic data only) on real choir recordings and find that in the absence of a large training set of real recordings, NMF still performs better than Wave-U-Net in this setting. To our knowledge, this paper is the first to study source separation of choral music.

1. INTRODUCTION

In this paper, we set out to investigate the application of source separation to choral music. We aim to take a recording of choral music and extract from it individual recordings for each of the choir sections (normally soprano, alto, tenor, and bass).

Audio source separation refers to extracting one or more sound sources of interest from a recording that involves multiple sound sources [1]. The musical applications of audio source separation include separating instruments in a recording and generating 'karaoke' tracks of songs by separating the accompaniment and the lead vocals [2]. To the best of our knowledge, this paper is the first to attempt separation of choral music. Separation of choral music enables applications such as fine-grained editing, analysis, and automatic creation of practice tracks (recordings of individual choir parts used by singers as an aid for learning new music) from professional choir recordings.

At the outset, choral music separation would seem a challenging task. Every choir section is composed of multiple singers singing simultaneously with slight variations in pitch and in timing, and every singer has a unique voice timbre. It follows that the resulting 'choral timbre' has extremely varied acoustical characteristics. Furthermore, an important goal in choral performance is achieving *blend* between singers, so that the choir is perceived by listeners as one coherent sound source [3]. This blend can naturally hinder the operation of an algorithm wishing to separate the choir. Choral music is often recorded in highly reverberant spaces such as churches, and the reverberations constitute yet another hurdle for separation. Finally, choirs are seldom recorded in a 'one voice per track' setting [4], and this lack of multi-track recordings makes it harder to design and validate source separation systems.

The rest of this paper is structured as follows. In Section 2 we review related work. In Section 3, we present a dataset of synthesized Bach chorale harmonizations. In Section 4, we establish baseline separation performance for choral music using NMF [5]. In Section 5, we apply a deep learning separation technique called Wave-U-Net [6] to choral music and in Section 6 we extend it to incorporate musical scores into the separation process. In Section 7, we present the results of several experiments conducted to determine the effectiveness of the proposed techniques.

2. RELATED WORK

Recently, the state of the art in source separation has advanced considerably, with some applications in speech even surpassing ideal time-frequency magnitude masking [7]. In music, one of the most common applications is vocals and accompaniment separation [8]. In this task, deep learning methods show the best performance among separation techniques [9]. Some state-of-the-art techniques operate on spectrograms [10,11] while others operate directly on the signal [6,7,12–14]. The reader is referred to [15] for a review of deep learning for speech separation and to [2,8] for overviews of music separation.

U-Net [16] is a prominent deep learning separation technique. Originally used for semantic segmentation of biomedical images [17], U-Net employs an encoder-decoder CNN architecture with skip connections to process the input on multiple scales. Wave-U-Net [6] extends

Attribution: Matan Gover, Philippe Depalle, "Score-Informed Source Separation of Choral Music", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

U-Net but instead of processing spectrograms it is applied directly to the signal. Demucs [12] is also based on a U-Net architecture and operates on the time-domain signal, with added features such as gated linear units and a recurrent layer between the encoder and the decoder.

2.1 Score-Informed Source Separation

The musical score, when available, is an invaluable source of detailed information on the mixture, such as instrumentation, pitch, and timing. Score-informed separation techniques use this information to guide the separation process [18]. One of the earliest techniques is synthesizing a signal from the target source's score and then using this signal as a reference [19–21]. Another technique is creating harmonicity-based masks or constraints driven by the note pitches and timings specified in the score [22–24]. Scores have also been used extensively as factorization constraints in the framework of NMF and its extensions [25–30].

More recently, scores have also been integrated into deep learning-based separation techniques. In [31], an autoencoder network was trained while imposing score-based constraints on the latent representation so that each latent unit represents a single note. Separation was then performed on a note-by-note basis. A technique for orchestral music separation [32] used a CNN that operates on 'score-filtered' spectrograms.

3. SYNTHESIZED BACH CHORALES DATASET

For training source separation techniques based on supervised learning, a large dataset of multi-track recordings is required. For example, the MUSDB18 dataset [9] for vocals and accompaniment separation contains 150 songs with a total duration of about 10 hours. Unfortunately, such a dataset of choir recordings does not currently exist. The Mixing Secrets dataset [1] contains some multi-microphone choral recordings, but there is significant leakage between the microphones. Choral Singing Dataset [33] is a good multi-track dataset, but it consists of only three songs.

In the absence of a large choral music dataset, we opt to use a synthesized dataset. Recently, a method for choir synthesis was proposed based on voice cloning [34], but unfortunately the implementation and the dataset are not publicly available. Choir audio tracks are often produced using commercial sample libraries [2] that contain thousands of professionally recorded choir samples. Unfortunately, these sample libraries are prohibitively expensive.

Previous work has shown that synthetic training data does not have to sound realistic for a model to generalize well [35, 36]. In light of this, we choose a relatively simple and cheap approach. We use the FluidSynth software synthesizer [37], which converts MIDI messages to audio by using audio samples and synthesis rules stored in a SoundFont file. We use the 'Choir Aahs' preset from the `MuseScore_General` SoundFont [3]. Each sample in this preset is a short recording of a single choir section singing a sustained note on an 'aah' vowel with a single pitch. To synthesize a pitch that does not have an associated sample, FluidSynth pitch-shifts the sample that has the closest pitch. To synthesize a note that is longer than the corresponding sample, a predefined segment of the sample is looped.

3.1 Bach Chorale Harmonizations

We construct our dataset from a well-known corpus of chorale harmonizations by J. S. Bach. A chorale is a Lutheran church hymn [38]. Bach harmonized around 400 chorales as part of large-scale vocal compositions as well as shorter works [39]. Bach's chorales are highly structured and this makes them good candidates to serve as a coherent dataset for source separation. They are written for four voices in homorhythmic texture [39]. The rhythm consists mainly of quarter notes and eighth notes. Structurally, the chorales are built as a sequence of short phrases, each ending with a fermata (musical pause).

3.2 Data Augmentation

Real-world choir recordings possess many sources of variability that are absent from Bach chorales. In order to make our dataset more closely resemble real-world recordings, we augment it with three added features: simulated breaths, random omitted notes, and tempo variations.

To simulate breaths between phrases, we insert a one-beat-long rest in all voices simultaneously every eight beats. To simulate sections in which one or more voices are silent while the other voices continue to sing, we randomly choose 10% of the notes in each voice and change them into rests. To add tempo variation, we synthesize each chorale at a random tempo between 70 and 100 BPM.

3.3 Synthesis Procedure

To synthesize our dataset we read the corpus of Bach chorales in MusicXML format using the music21 library [40]. From the 371 chorales in the Riemenschneider edition we exclude 20 chorales that contain instrumental parts or more than four vocal parts. The 351 remaining chorales are split into three partitions: training (270 chorales), validation (50), and test (31). For each chorale we export four MIDI files, one file per voice, and synthesize them using FluidSynth. [4] The total duration of the dataset is 3h 48m.

4. BASELINE: SCORE-INFORMED NMF

We establish a baseline for separation performance on our dataset using a classic separation technique: non-negative matrix factorization (NMF) [5, 41]. NMF factorizes a mixture spectrogram into two matrices: basis signals and temporal activations. To constrain the NMF separation process we use a score-based initialization scheme for the basis

[1] `http://www.cambridge-mt.com/ms/mtk/`

[2] e.g., `http://soundsonline.com/hollywood-choirs`

[3] `https://bit.ly/musescore-general`

[4] The code to generate the dataset is available at: `https://github.com/matangover/synthesize-chorales`

signals and activations matrices [25]. We dub this technique SI-NMF. Our implementation is available online.[5] We use the variant dubbed I_{WH} in the original paper, which imposes constraints on both basis signals and activations. For the STFT we use a Hann window with a size of 2,048 samples (the dataset sampling rate is 22,050 Hz). The SI-NMF score-based constraints allow some tolerance to account for slight pitch and timing variations in the mixture. Since in our dataset the scores are perfectly aligned to the mixture, we use onset tolerance of 0 and offset tolerance of 0.2 seconds (to account for note decay). We use pitch tolerance of 1 semitone. These parameters were found to give the best results after comparing several alternatives.

5. WAVE-U-NET FOR CHORAL MUSIC

To improve on the SI-NMF baseline, we propose to apply Wave-U-Net (described in Section 2). Wave-U-Net attained good results in the SiSEC 2018 evaluation campaign [9] and its code is publicly available. Since Wave-U-Net operates in the time domain, it may be well suited for separating sources with overlapping partials, which are ubiquitous in choral music and may pose a challenge for methods that rely on spectrogram masking [12].

We follow the training procedure used in the original Wave-U-Net paper. Every training batch consists of 16 short (6-second) segments extracted from the training set at random positions. The Adam optimizer [42] is used with the mean squared error loss and an initial learning rate of 0.0001. The validation set is used for early stopping.

In the original implementation of Wave-U-Net, a single model is trained to extract all sources at the same time. This is economical in terms of model weights and training time, but it forces the latent representations to be generic enough to fit all sources. Instead, we propose to train a separate model for each extracted source. This way the model can be specifically geared to extract each of the sources.

6. SCORE-INFORMED WAVE-U-NET

We propose to condition Wave-U-Net on the musical score of the separated sources to improve separation quality.[6] The pitch and timing information contained in the score can help overcome the challenges of separating choral music. Timbre is generally a useful differentiating factor for separation, but the timbres of the women's voices (soprano and alto) are similar to each other, and so are the men's (tenor and bass). Relying on the pitch range of each choir part is also not sufficient for separation because the ranges have considerable overlap. For example, an F4 note (F above middle C) could easily be sung by the soprano, alto, or tenor, and in rare cases also by the bass [3, p. 234]. The standard SATB (soprano-alto-tenor-bass) ordering of the voices could sometimes be used for separation, but this ordering is not always kept, and in any case it could only be used in sections where all voices sing at the same time. Hence, in many cases the musical score may be the only way to associate notes to a specific voice in choral music.

6.1 Score Representations

Our dataset provides the score for each part as a monophonic MIDI file indicating each note's onset time, offset time, and pitch. We transform the MIDI note sequence into a representation that can be efficiently processed by Wave-U-Net. In choral music, every part sings at most one note at a time. (In the case of *divisi*, such as when soprano is split into soprano 1 and soprano 2, we can treat every divisi section as a distinct source.) Therefore, we represent a part's score as a time series that indicates the active pitch (if any) at every time point. To keep the score aligned with the network's audio input, we use the same sampling rate for the audio and the score representation. We investigate four different score representations [43].

normalized pitch. A part's score is represented as a vector in which every element indicates the active pitch at the corresponding time instant. Since the range of MIDI note numbers (0 to 127) is radically different from the range of the audio input (-1 to 1), we normalize the note number to the range $[0, 1]$, and use the special value -1 to indicate no note is active. Given a MIDI note number M, the normalized pitch S_{n} is computed as:

$$S_{\text{n}}(M) = \frac{M - M_{\text{min}}}{M_{\text{max}} - M_{\text{min}}}, \tag{1}$$

where M_{min} and M_{max} are the minimum and maximum expected note pitches, respectively. We set $M_{\text{min}} = 36$ and $M_{\text{max}} = 84$, based on the normal choral voice ranges: from C2 (very low bass note) to C6 (very high soprano note).

pitch and amplitude. In order to better encode the difference between sung notes and silence, we introduce a two-channel representation, in which one channel represents pitch and the other represents amplitude. The pitch channel S_{p} is normalized to the range $[-1, 1]$, as given by: $S_{\text{p}}(M) = 2S_{\text{n}}(M) - 1$. The amplitude channel is boolean: its value is 1 when a note is active and 0 otherwise. When no note is active the pitch channel is set to -1.

piano roll. The score is represented as a one-hot matrix of size $p \times n$ where p is the number of available pitches ($p = M_{\text{max}} - M_{\text{min}} + 1$) and n is the length of the network's audio input. The matrix element at row p_i and column n_j is set to 1 if a note with pitch p_i is active at time n_j. Otherwise, the element is set to 0.

pure tone. Since the model inputs are audio, we propose to represent the score in a simplistic audio-like form. We use a pure tone signal constructed as a piecewise sine function where the frequency is controlled by the active note's pitch. For simplicity, we do not create smooth note transitions, so any note onset will result in a discontinuity. The pure tone frequency f is determined by the standard MIDI note number to frequency mapping:

$$f(M) = 440 \cdot 2^{\frac{M-69}{12}}. \tag{2}$$

When there is no active note, f is set to 0. The score vector

[5] https://git.io/si-nmf

[6] https://git.io/si-Wave-U-Net

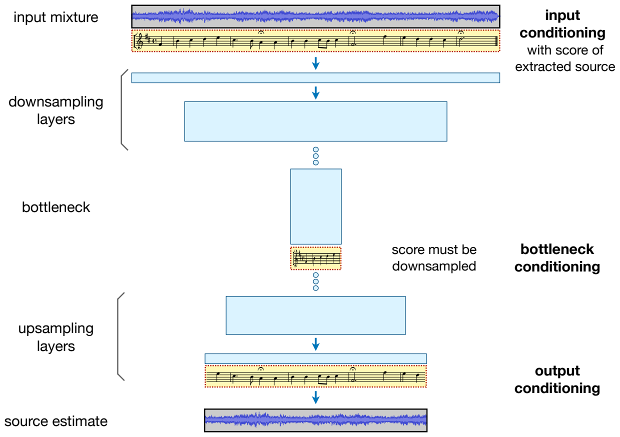

Figure 1. Conditioning locations for Wave-U-Net (showing a model that extracts a single source, see Section 5)

then receives the following value at each sample index i:

$$S_{\mathrm{t}}(M, i) = \sin\left(2\pi f(M) * \frac{i}{F_s}\right), \quad (3)$$

where F_s is the sample rate of the model's audio input.

All four score representations do not differentiate between a sustained note and a repeated note: consecutive notes with the same pitch are represented the same as one note with a longer duration. Devising a score representation that does encode this difference is left to future work.

6.2 Conditioning Method

Common methods for conditioning neural networks include *concatenation*, in which a conditioning tensor is concatenated to the input tensor; *biasing*, in which a conditioning tensor is added to the input tensor; and *scaling*, in which the input tensor is multiplied element-wise by a conditioning tensor [44]. In this work we use concatenation, which is equivalent to biasing with a linear transformation applied to the conditioning [44].

We investigate three conditioning locations in the Wave-U-Net architecture (see Figure 1): *input conditioning* (score is concatenated to the input audio before the decoder), *output conditioning* (score is concatenated to the decoder's output before the output layer), and *input-output conditioning* (a combination of both). Other conditioning locations are also possible, but they would require a transformed score representation. Conditioning at the bottleneck, for example, would necessitate resampling the score information to the bottleneck's much lower temporal resolution, thus discarding important timing information contained in the score. Conditioning at the bottleneck could work well when the conditioning has no temporal dimension, such as instrument labels [45].

6.3 Multi-Source Training

In addition to the standard method of training the network to extract specific voices, we propose a *multi-source* model variant which can separate any one of the four voices given only that voice's score. To achieve this, we train a model to extract a single voice from the mixture, where every training example consists of a mixture segment (used as input to the model), the score of one random voice out of the four voices (used to condition the model), and the corresponding audio for that voice as the target to extract (used to compute the loss). Since training examples do not explicitly specify which voice they correspond to, the model learns to extract the desired voice based on its score alone. Whereas a normal score-informed model *could* use the score to improve separation results, the multi-source model *must* make use of the score. In this sense, the separation is not only score-informed, it is *score-guided*. A multi-source model also gives greater flexibility by allowing users to choose individual notes to extract, possibly alternating between voices. Furthermore, multi-source training can enable a model trained only on four-voice mixtures to be used on recordings with any number of voices.

7. EXPERIMENTS AND RESULTS

To evaluate model performance, we use the SDR metric [46] provided by the BSS Eval library (version 4) [9] with its default settings [7]. Like SiSEC 2018 and subsequent works, we report median SDR rather than mean in order to reduce the effect of outliers. We compare all proposed model variants in 6 experiments, listed in Table 1. Audio examples are available online. [8] We assess whether certain methods perform better than others by reporting p-values from pairwise Conover–Iman tests [47] (also used by [9]; we adjust for multiple comparisons using the Bonferroni method [48]), always after rejecting the Kruskal–Wallis [49] null hypothesis with $P < 0.001$.

Experiment	Method	Score-Informed	Model Type
1	SI-NMF	yes	-
2	Wave-U-Net	no	one model for all voices
3	Wave-U-Net	no	one model per voice
4	Wave-U-Net	yes	one model for all voices
5	Wave-U-Net	yes	one model per voice
6	Wave-U-Net	yes	one model: multi-source

Table 1. List of experiments

7.1 Experiments 1–3: SI-NMF and Wave-U-Net

A comparison of separation performance of SI-NMF and Wave-U-Net on the test set is shown in Figure 2. While SI-NMF achieves decent results, Wave-U-Net consistently outperforms it in all voices by a large margin ($P < 0.001$).

In SI-NMF, interferences between estimated sources are very low due to the hard constraints imposed using the score. However, estimated sources contain noticeable amplitude modulation artifacts. These are likely caused by the use of static spectral templates, which cannot effectively model the continuous evolution of spectral parameters in choral music. Source-filter signal models can be integrated into NMF to improve its performance in such cases [50–52]. The effectiveness of such models for choral

[7] We also provide supplementary SIR, SAR, and ISR evaluations. [8]

[8] https://www.matangover.com/choirsep-ismir

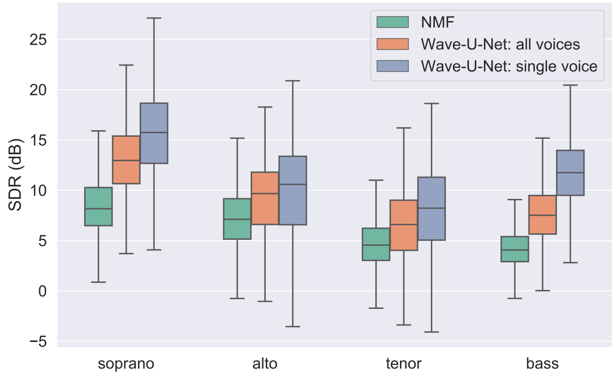

Figure 2. Results for SI-NMF and Wave-U-Net

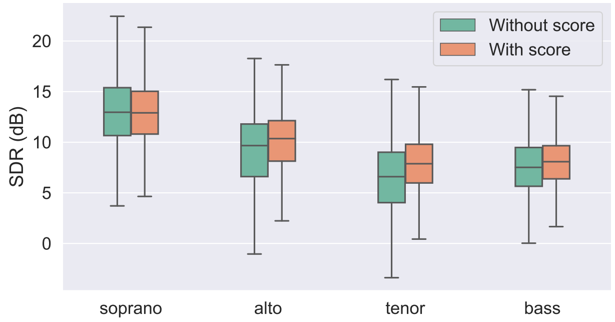

Figure 3. Results from Experiment 4 (with score; score type: normalized pitch, conditioning location: input) compared to Experiment 2 (without score)

music may be limited, however, as a choir section is actually composed of multiple sound sources (singers). Figure 2 further shows that single-voice Wave-U-Net (Experiment 3) is superior to all-voice Wave-U-Net (Experiment 2) ($P < 0.001$).

Examination of segments in which the model achieved a particularly low SDR reveals that the most common source of errors is misclassified notes (that is, when the model assigns notes to the wrong voice) [43]. One cause for misclassified notes is voice crossings, which occur when the normal voice ordering is violated. The fact that voice crossings cause misclassification shows that the model has learned to rely on the standard ordering of the voices. Misclassified notes also occur in segments in which one voice is silent while the other voices continue to sing. In such segments the model cannot always infer which voice is silent due to the overlap between voice ranges.

7.2 Experiment 4: Score-Informed, Extract All Voices

In Experiment 4 we examine the effect of adding score conditioning to the model from Experiment 2. We train 12 score-informed model variants: all combinations of 4 score representations and 3 conditioning locations. Figure 3 shows that adding the score improves median SDR in all voices ($P < 0.001$) except for soprano ($P > 0.05$).

Figure 4 compares all score conditioning methods. Conditioning location has no consistent effect on soprano and bass separation. For alto and tenor, however, output conditioning is overall worse than both input and input-output conditioning ($P < 0.001$). We suspect that output conditioning performs poorly because the Wave-U-Net output layer is a simple sample by sample dot-product (convolutional layer with kernel of size 1).

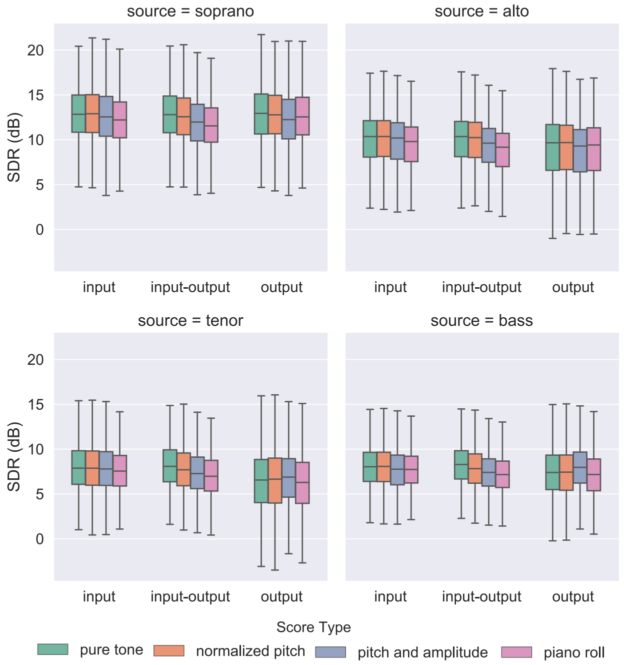

Figure 4. Results from Experiment 4 by voice, score type, and conditioning location

7.3 Experiment 5: Score-Informed, Extract Single

Figure 5 compares the performance of score conditioning methods for tenor extraction in Experiment 5. We compare results for tenor specifically because it is the most challenging to separate (it achieved the lowest median SDR in most experiments). Output conditioning gives the worst performance and has no significant effect compared to no score at all ($P > 0.05$). It appears the models conditioned at the output have learned to simply ignore the score. For input and input-output conditioning, the choice of score type has no effect, and all score types perform considerably better than no score at all ($P < 0.001$), with an improvement of up to 2.7 dB in median SDR.

7.4 Experiment 6: Score-Informed, Multi-Source

This experiment tested the effect of score conditioning method on multi-source training (described in Section 6.3). We do not include a figure due to limited space, see website [8] for results. Models using output conditioning perform very poorly, confirming the results of Experiments 4 and 5. Other than that, conditioning method does not have an effect in this experiment. The difference in median SDR between the best method (pitch and amplitude, input) and the worst method (piano roll, input-output) is only 0.4 dB.

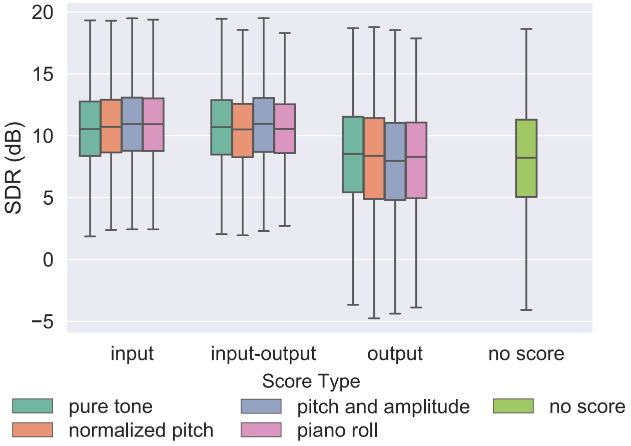

Figure 5. Comparison of score conditioning methods in Experiment 5 (on tenor only), with the non-score-informed counterpart (Experiment 3) shown for reference

7.5 Overall Comparison

In Figure 6 we compare results from all experiments. For the score-informed models we use the conditioning method that performed best, taking into account all experiments and all voices (score type: pitch and amplitude, conditioning location: input). As expected, using the score improves performance mainly for the inner voices (alto and tenor), as they are more prone to induce misclassified notes due to voice crossings and vocal range overlap (see Section 7.1). Examination of frames with misclassified notes confirms that using the score eliminates this problem [43, p. 95].

The score-informed single-source model has the best performance overall. For alto and tenor, this model achieves a 2.7 dB improvement in median SDR compared to the best non-score-informed model ($P < 0.001$). For soprano, the improvement is only 0.5 dB ($P < 0.001$) and for bass performance is degraded by 0.06 SDR ($P < 0.01$). Compared to the NMF baseline, score-informed Wave-U-Net improves median SDR by 6.2 to 8.1 dB ($P < 0.001$).

Interestingly, for tenor and alto, the multi-source model outperforms the non-score-informed single-source model ($P < 0.001$), even though the multi-source model uses only a quarter of the parameters (because it uses a single model for all four voices).

Listening to audio results of score-informed models, [8] we notice that most score conditioning methods result in audible clicks at note boundaries. This is likely caused by the discontinuity of the score representations at these locations. These clicks hardly affect the SDR evaluations because they are highly localized. Using the pure tone score representation eliminates these clicks almost completely.

7.6 Evaluation on Real-World Recordings

Although our models have only been trained on synthesized data, we also evaluate using real choir recordings from the Choral Singing Dataset [33]. In this evaluation, non-score-informed Wave-U-Net (Experiment 3 model) performs poorly with a median SDR of 0 dB (for all voices combined). Score-informed Wave-U-Net performs better with SDR of 1.4 and 1.5 dB (models from Experiments 5 and 6, respectively). SI-NMF outperforms Wave-U-Net by a large margin with SDR of 5.6 dB ($P < 0.001$).

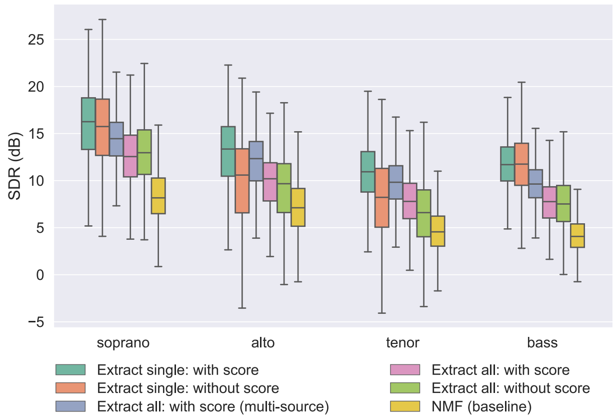

Figure 6. Comparison of results from all experiments

Listening to estimated sources we notice that score-informed Wave-U-Net predicts all the right notes, but cannot faithfully generate the lyrics and unique timbre of the specific choir, likely due to it being trained on a dataset containing only a single choir without any lyrics. SI-NMF predictions also omit many of the lyrics and timbre variations, but are nonetheless better than Wave-U-Net in this case. This shows that to be effective on real-world recordings, Wave-U-Net needs to be trained on a more representative dataset. We postulate that if score-informed Wave-U-Net (or similar methods) could be trained on a diverse dataset of choral recordings, it would achieve an improvement over SI-NMF that is comparable to the improvement that it has achieved on the synthesized dataset.

8. CONCLUSIONS

In this paper we investigated source separation of choral music. Due to the lack of publicly available datasets, we developed a dataset of synthesized Bach chorales. We established baseline separation performance using a score-informed NMF method. We then showed that NMF is outperformed by Wave-U-Net, a deep learning separation technique. We further proposed to condition Wave-U-Net on musical scores. Our experiments with several conditioning methods showed that using the score improves separation quality. We introduced multi-source training, in which a single model separates any of the four choir voices using only the score as a guide. We found that multi-source training performs comparably to single-source training, even though it requires much less resources.

When evaluated on real choir recordings, SI-NMF still outperforms Wave-U-Net. Hence, a major challenge that remains is compiling a multi-track choir recording dataset to be used for training. Until such a dataset is available, better choir synthesis methods could be used. Another avenue for improvement would be to consider more versatile conditioning methods, such as FiLM layers [53, 54].

9. ACKNOWLEDGEMENTS

This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) awarded to PD (RGPIN-2018-05662). MG would like to thank the Jewish Community Foundation of Montreal (Jenny Panitch Beckow Memorial Scholarship) and the Schulich School of Music at McGill University for providing funding. We thank Compute Canada for providing computing resources used in this work.

10. REFERENCES

[1] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley & Sons, 2018.

[2] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, "Musical source separation: An introduction," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, 2019.

[3] B. Smith and R. T. Sataloff, *Choral Pedagogy*, 3rd ed. Plural Publishing, 2013.

[4] K. Ihalainen, "Methods of choir recording for an audio engineer," Bachelor's thesis, Tampere University of Applied Sciences, 2008.

[5] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[6] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. of the International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 334–340.

[7] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2018.

[8] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, 2018.

[9] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *14th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Guildford, UK, 2018, pp. 293–305.

[10] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - a reference implementation for music source separation," *Journal of Open Source Software*, 2019. [Online]. Available: https://doi.org/10.21105/joss.01667

[11] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MM-DenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Tokyo, Japan, 2018, pp. 106–110.

[12] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[13] F. Lluís, J. Pons, and X. Serra, "End-to-end music source separation: Is it possible in the waveform domain?" *arXiv:1810.12187 [cs, eess]*, 2018.

[14] E. M. Grais, D. Ward, and M. D. Plumbley, "Raw multi-channel audio source separation using multi-resolution convolutional auto-encoders," in *Proc. of the 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, pp. 1577–1581.

[15] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[16] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proc. of the International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Cham, Switzerland, 2015, pp. 234–241.

[18] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, 2014.

[19] Y. Meron and K. Hirose, "Separation of singing and piano sounds," in *5th International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[20] C. Raphael, "A classifier-based approach to score-guided source separation of musical audio," *Computer Music Journal*, vol. 32, no. 1, pp. 51–59, 2008.

[21] J. Ganseman, G. J. Mysore, J. S. Abel, and P. Scheunders, "Source separation by score synthesis," in *Proc. of the International Computer Music Conference (ICMC)*, New York, NY, 2010, pp. 462–465.

[22] A. Ben-Shalom and S. Dubnov, "Optimal filtering of an instrument sound in a mixed recording given approximate pitch prior," in *Proc. of the International Computer Music Conference (ICMC)*, San Francisco, CA, 2004.

[23] Y. Li, J. Woodruff, and D. Wang, "Monaural musical sound separation based on pitch and common amplitude modulation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 7, pp. 1361–1371, 2009.

[24] Z. Duan and B. Pardo, "Soundprism: An online system for score-informed source separation of music audio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1205–1215, 2011.

[25] S. Ewert and M. Müller, "Using score-informed constraints for NMF-based source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 129–132.

[26] R. Hennequin, B. David, and R. Badeau, "Score informed audio source separation using a parametric model of non-negative spectrogram," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 45–48.

[27] U. Şimşekli and A. T. Cemgil, "Score guided musical source separation using Generalized Coupled Tensor Factorization," in *Proc. of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, 2012, pp. 2639–2643.

[28] F. J. Rodriguez-Serrano, Z. Duan, P. Vera-Candeas, B. Pardo, and J. J. Carabias-Orti, "Online score-informed source separation with adaptive instrument models," *Journal of New Music Research*, vol. 44, no. 2, pp. 83–96, 2015.

[29] F. J. Rodriguez-Serrano, S. Ewert, P. Vera-Candeas, and M. Sandler, "A score-informed shift-invariant extension of complex matrix factorization for improving the separation of overlapped partials in music recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 61–65.

[30] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *Journal of Electrical and Computer Engineering*, vol. 2016, 2016.

[31] S. Ewert and M. B. Sandler, "Structured dropout for weak label and multi-instance learning and its application to score-informed source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 2277–2281.

[32] M. Miron, J. Janer, and E. Gómez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *Proc. of the International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 55–62.

[33] H. Cuesta, E. Gómez, A. Martorell, and F. Loáiciga, "Analysis of intonation in unison choir singing," in *Proc. of the International Conference on Music Perception and Cognition (ICMPC)*, Graz, Austria, 2018.

[34] M. Blaauw, J. Bonada, and R. Daido, "Data efficient voice cloning for neural singing synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 6840–6844.

[35] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in *Proc. of the International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 71–78.

[36] M. Miron, J. Janer, and E. Gómez, "Generating data to train convolutional neural networks for classical music source separation," in *Proc. of the Sound and Music Computing Conference*, Espoo, Finland, 2017, pp. 227–233.

[37] D. Henningsson and F. D. Team, "FluidSynth real-time and thread safety challenges," in *Proc. of the 9th International Linux Audio Conference*, Maynooth, Ireland, 2011, pp. 123–128.

[38] R. L. Marshall and R. A. Leaver, "Chorale," in *Grove Music Online*. Oxford, UK: Oxford University Press, 2001.

[39] ——, "Chorale settings," in *Grove Music Online*. Oxford, UK: Oxford University Press, 2001.

[40] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. of the International Society for Music Information Retrieval Conference*, Utrecht, Netherlands, 2010, pp. 637–642.

[41] A. Cichocki, R. Zdunek, and S.-i. Amari, "New algorithms for non-negative matrix factorization in applications to blind source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, Toulouse, France, 2006, pp. 621–624.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2014.

[43] M. Gover, "Score-informed source separation of choral music," Master's thesis, McGill University, 2019.

[44] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. de Vries, A. Courville, and Y. Bengio, "Feature-wise transformations," *Distill*, vol. 3, no. 7, p. e11, 2018.

[45] O. Slizovskaia, L. Kim, G. Haro, and E. Gómez, "End-to-end sound source separation conditioned on instrument labels," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 306–310.

[46] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[47] W. J. Conover and R. L. Iman, "On multiple-comparisons procedures," Los Alamos Scientific Laboratory, Tech. Rep. LA-7677-MS, 1979.

[48] C. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilità," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[49] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[50] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *Proc. of the International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009, pp. 327–332.

[51] J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David, "Main instrument separation from stereophonic audio signals using a source/filter model," in *17th European Signal Processing Conference*, Glasgow, UK, 2009, pp. 15–19.

[52] T. Nakamura and H. Kameoka, "Shifted and convolutive source-filter non-negative matrix factorization for monaural audio source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 489–493.

[53] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *32nd Conference on Artificial Intelligence (AAAI-18)*, New Orleans, LA, 2018.

[54] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations," in *Proc. of the International Society for Music Information Retrieval Conference*, Delft, Netherlands, 2019.