Source Separation Methods for Computer-assisted Orchestration

Luke Dzwonczyk
Center for New Music and Audio Technologies (CNMAT)
University of California, Berkeley
Berkeley, CA
dz.luke@berkeley.edu

Léo Chédin
École Normale Supérieure Paris-Saclay
Université Paris-Saclay
Paris, France

Alejandro Saldarriaga-Fuertes
CNMAT
UC Berkeley
Berkeley, CA

Max Sherr
CNMAT
UC Berkeley
Berkeley, CA

Hélène-Camille Crayencour
Laboratoire des Signaux et Systèmes
CentraleSupélec, CNRS, Université Paris-Saclay
Paris, France

Carmine-Emanuele Cella
CNMAT
UC Berkeley
Berkeley, CA

Abstract

In this paper, we study the possibility of adding source separation as a pre-processing step to the computer-assisted orchestration process. We first discuss the motivation of this addition and its potential to increase the quality of orchestrations of multi-layered sounds. Second, we select several state-of-the-art models for both music source separation (separation of instruments) and universal sound separation (separation of arbitrary sounds), and compare their effectiveness for the task of orchestration. We assess which methods best suit the needs of orchestration by applying them on our own target sounds, orchestrating the separated outputs, and finally comparing them to the orchestration of the same target without separation. Our experiments show that source separation improves the quality of orchestrations, and the more accurate the separation, the better the resulting orchestration. Finally, we compare unsupervised methods to supervised methods for separation, and comment on the effect of training data selection on performance of supervised methods.

1 Introduction

Computer-assisted composition is a field that focuses on the creation of computational tools to aid in the musical composition process (Fernandez and Vico, 2013; Ariza, 2005). Within this field lies target-based computer-assisted musical orchestration, a method which uses orchestral instruments to recreate the timbre and temporal evolution of a target sound. It is the process of creating an orchestral score that best matches an arbitrary target sound given a similarity metric and constraints (Maresz, 2013). In other words, finding which instruments playing which notes sounds the most similar to the given sound. A goal of computer-assisted orchestration is to help composers by accelerating certain creative processes. In particular, target-based computer-assisted orchestration helps composers explore new timbral possibilities by combining instrumental samples in such a way as to mimic the timbre of a given target sound.

Target-based computer-assisted orchestration is a complex problem because it attempts to recreate the timbre of different sounds, and timbre is a difficult parameter to model. The task becomes more complicated when you consider multi-layered sounds that are a combination of multiple sources.

3rd Conference on AI Music Creativity (AIMC 2022)

When two or more sources overlap in time, the orchestration algorithm cannot distinguish between them. As such, the orchestration is rigid: the whole orchestra suddenly changes configuration when a new source dominates, without any blending with the previous dominating source.

Our hypothesis is that if source separation is applied as a pre-processing step, then the separated layers of a sound can be independently orchestrated and then recombined, resulting in an improved orchestration. This logic follows from the orchestration principle of *dovetailing* (Adler, 2016, p. 467-473). Dovetailing is a well known practice aimed at creating a blend of the instruments used during orchestration in order to transition between timbres. Technically, dovetailing is the swapping and overlapping of musical lines between different instruments: musical lines are scattered across multiple instruments, connected with overlapping pitches

Orchidea is a framework and set of tools that is currently considered the state-of-the-art for computer-assisted orchestration. It embeds the target in a feature space and uses a jointly-optimized heuristic and constraint solver to find a combination of samples that best match the target (Cella and Esling, 2018; Cella, 2020). All of our orchestrations are done using Orchidea's command line tools for batch processing, which is available for download at `http://www.orch-idea.org`.

Orchidea is capable of performing both static and dynamic orchestration. In static orchestration, temporal evolution is ignored and the target is represented as a single vector of timbral features. The output is a single onset of notes that does not change over time. In contrast, dynamic orchestration considers a time series of features, which allows the orchestration to evolve over time and contain multiple onsets of notes. Orchidea is able to perform dynamic orchestration by cutting the target in time into multiple segments, and then orchestrating each segment. In this paper, we perform only static orchestrations in order to better observe the effects of source separation. The reason for this is that we wish to disentangle the two difficult problems of source separation and time segmentation. By only performing static orchestrations, we reduce the complexity of the problem by removing time segmentation and can better observe the effects of source separation.

The paper is organized as follows: in section 2 we introduce sound source separation and list other musical tasks in which source separation has had success. Section 3 will discuss our methods and walk through the testing procedures we employed in our experiments. Sections 4 and 5 will describe the findings of our experiments, an analysis of the results, and offer possibilities for future work.

The code for this paper can be found at: `https://github.com/dzluke/AIMC2022`, and you can listen to a selection of targets and orchestrations here: `https://dzluke.github.io/AIMC2022/`.

2 Background

Sound source separation is the process by which a single audio file is separated into multiple sound sources. A perfect separation is able to dissect a sound exactly into its constituent parts, in which each part is an independent sonic event. Music source separation is a specific application of source separation in which the input audio is music that is comprised of a subset of instruments or sounds. For example, one music source separation method could attempt to split a song into three parts: voice, bass, and drums. Another aimed at orchestral music may try to separate the input into families of instruments: woodwinds, brass, strings, and percussion. In contrast, universal sound separation does not operate under the assumption that the input is of a musical nature, and can be applied to arbitrary sounds. In this case, the number of sources expected is specified, and the method will attempt to divide the input into the given number of sources.

Using source separation techniques as a pre-processing step has been shown to improve results in several music processing tasks. Source separation can be used to remove part of the spectrum in order to enhance the clarity of the features of interest for the task at hand. For instance, in the task of automatic chord estimation from audio, some authors have chosen to remove part of the spectrum related to percussive sounds to improve chord accuracy (Reed et al., 2009). Similarly some authors have investigated how beat tracking can be improved by using source separation as a pre-processing step for difficult songs with highly expressive vocals (Zapata and Gómez, 2012). Source separation can also be applied to separate different components of the signal and work on them separately to reduce the complexity of the task. For instance, in the context of tempo detection in Chordia and Rae (2009), the authors decompose the signal into sources to reduce rhythmic complexity, under the idea that some layers may be more rhythmically regular than the overall mix. Another example can be

found in Gómez et al. (2018) where harmonic/percussive and solo/accompaniment source separation techniques are investigated as a pre-processing step to improve predominant instrument recognition in jazz music.

Consider a motivating example: a target that has a continuous drone sound throughout and a melody playing above this drone. Without source separation, Orchidea would detect each note of the melody as a new onset and cut the target in time at each new note. However, this segmentation in time would also affect the drone, cutting it in time and forcing it to have a new onset each time the melody changes notes. The resulting orchestration would have multiple onsets for the drone, even though only one onset is needed at the beginning of the drone. When source separation is applied to this target, the drone and melody could be orchestrated separately. Therefore, the drone could be orchestrated by a single onset that lasts for the duration of the drone. At the same time, the melody could have as many new onsets as needed without affecting the drone. The orchestration would be greatly improved by removing the unnecessary onsets in the orchestration of the drone.

3 Experimental Methodology

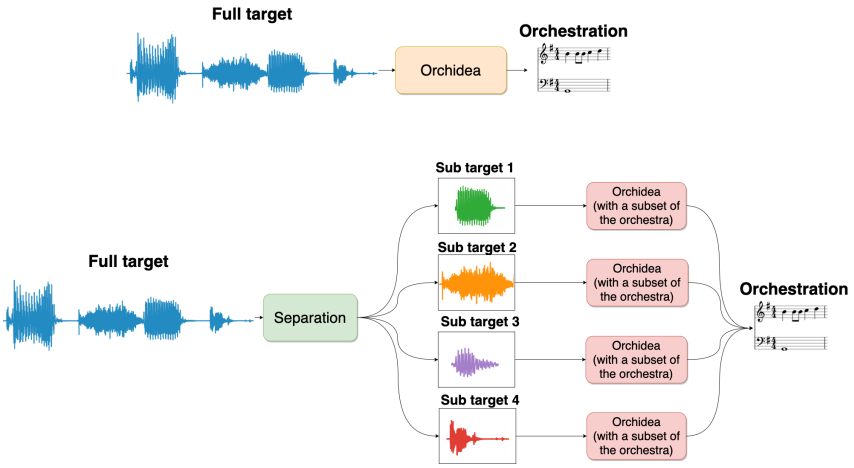

Figure 1: Diagram comparing the process of orchestration with and without separation. At the top, the full target orchestration is created simply by orchestrating the target. At the bottom, the separation process is performed, and the resulting sub-targets are individually orchestrated with subsets of the orchestra before being recombined to obtain the separated orchestration.

We compare the effectiveness of different source separation methods for the task of orchestration by applying the separation methods to various targets, orchestrating the separated output of the method, and finally comparing this to the orchestration of the target if no separation was performed. This process is described in Fig. 1. Consider a target sound that is a combination of a low droning sound and high-pitched whistle. With separation, these two sounds would be disentangled into two separate sub-targets and each would be individually orchestrated. The separated orchestrations would then be combined to create the solution (Fig. 1, bottom). This would then be compared to the orchestration if no separation had been performed (Fig. 1, top).

We test the effectiveness of five different supervised source separation methods on 300 custom made targets that are combinations of sounds from different databases. During testing, a target is input to a source separation method, which outputs four sub-targets. The accuracy of the separation is measured using the cosine distance. Then each sub-target is independently orchestrated with a randomly assigned subset of the full orchestra. These orchestrations are then combined to play simultaneously, creating a final orchestrated solution. Finally the distance between the target and solution is calculated, giving us a metric to compare the various separation methods. See Fig. 2 for a diagram of this process.

For a given target, the same split of the orchestra is used to orchestrate the separations. We randomly create these subsets of the orchestra in order to avoid any biases that could come from a hand-picking the instruments used to orchestrate each sub-target.

3.1 Data

We created 300 targets where each target was a combination of four source sounds. The sources come from the NIGENS (Trowitzsch et al., 2019) and BBC (BBC, 2021) databases, and freesound.org (Font et al., 2013). We selected a total of 90 samples from these databases, choosing sounds that fit the following criteria:

1. Static sounds that do not change harmonically over time
2. Sounds in which there is at least some pitched content and not only noise

The sounds chosen include alarms, bells, engine noises, soundscapes, synthesizer chords, and sound effects. Each target that we used for testing was a combination of four randomly chosen source sounds from the group of 90 sounds. When a target is created, the longest of the four sources is chosen to start playing at the beginning of the target. The other three sources are then randomly assigned different times to begin playing such that they will start between the beginning of the target and half-way through the longest source sound. This is done to ensure that the sources overlap and to minimize the amount of time in which there is only a single source sounding.

3.2 Separation methods

We consider source separation models trained to tackle two distinct separation problems: music source separation and universal sound separation. The music source separation models we test are Open-Unmix, Demucs, and Conv-Tasnet. The universal source separation models we test are non-negative matrix factorization (NMF) and TDCN++. Open-Unmix, Demucs, Conv-Tasnet, and TDCN++ are supervised models, and NMF is an unsupervised method.

It is important here to make the distinction that we did not train the supervised models ourselves. Therefore, we cannot guarantee that the data the models were trained on is optimal for solving our problem. However, one of the goals of this paper is to explore whether existing, pre-trained source separation models can separate non-musical sounds in a way that improves orchestration. We choose to include music source separation methods because of the diversity of different models available and the results these models achieve at their task.

3.2.1 Music source separation

The music source separation models we use are Open-Unmix (Stöter et al., 2019), Demucs (Défossez et al., 2019), and Conv-Tasnet (Luo and Mesgarani, 2018). For these architectures, we use models that are pre-trained on the MUSDB18 dataset (Rafii et al., 2017). The models output 4 stereo tracks that correspond to instrumental categories defined in the SiSEC 2018 Mus evaluation campaign: *vocals, drums, bass* and *other* (Stöter et al., 2018).

Open-Unmix (Stöter et al., 2019) is an open source implementation of a deep music source separation model, based on Uhlich et al. (2017). It is actually composed of multiple models that are trained for each instrumental target. Each of these models is trained on the specific target using the same architecture, based on a three layer bidirectional LSTM network. Open-Unmix operates on the Short Time Fourier Transform (STFT) of the input mixture. It predicts the target by multiplying the output of the LSTM network with the magnitude spectrogram of the input, essentially applying a mask, and uses Wiener filtering as a last processing step.

The **Demucs** (Défossez et al., 2019) architecture is a deep learning model for musical source separation. This models in the waveform domain, taking the stereo mixture as input, and outputting 4 stereo waveforms corresponding to the four categories of the MUSDB18 dataset. The Demucs models consists of a 6-layer convolutional encoder, two bidirectional-LSTM layers, and a 6-layer decoder made of transposed convolution. Skip U-net connections (Ronneberger et al., 2015) link encoder layers with their corresponding decoder layers. This model obtained state-of-the-art results on the MusDB set, surpassing Open-Unmix. We use the second version of Demucs which uses only the waveform, the third version being a hybrid model using both the waveform and a spectrogram.

We also used the **Conv-Tasnet** (Luo and Mesgarani, 2018) implementation of Défossez et al. (2019), who adapted the architecture, originally designed for speech separation at 8 kHz, for the task of music source separation. Conv-Tasnet is a deep learning model that is also based on three processing stages:

encoder, separation and decoder. The encoder transforms the input waveform into a high-dimensional feature representation that is optimized to separate different sources. Then, the separation step computes a multiplicative mask for each of the target sources. Finally, the decoder reconstructs the original waveforms using the masked features.

3.2.2 Universal sound separation

The second type of method we tested are the universal sound separation methods TDCN++ and NMF, which can separate arbitrary sounds, not just musical instruments. This method more closely matches our problem of separating non-musical target sounds.

Non-negative matrix factorization (NMF) (Cichocki and Phan, 2009; Févotte and Idier, 2011) is an unsupervised learning method, which takes an input non-negative matrix, in this case a spectrogram S, in $\mathbb{R}^{N \times M}$ and decomposes it into a basis matrix A in $\mathbb{R}^{N \times J}$ and a component matrix X in $\mathbb{R}^{J \times M}$ such that $S \approx A \times X$. We use the coordinate descent algorithm from the scikit-learn Python package based on Cichocki and Phan (2009) with J set to 4, which selects for 4 features in the dataset. We then output 4 mono waveforms containing each isolated feature F_i, where $F_i = A_i \times X_i$ for $i = 1, 2, 3,$ or 4.

TDCN++ is a model proposed by Kavalerov et al. (2019), based on Conv-Tasnet. It was modified in order to be used for general sound separation, and not only for speech. The main architecture remains the same as in Luo and Mesgarani (2018), and the changes only affect the masking network. More specifically, feature-wise normalization replaces global normalization, residual connection ranges are extended to improve the flow of information, and learnable scale parameters are introduced to weight the outputs of each residual layer. Those modifications help the model to better extract features without drastically increasing its size. We use a trained version of this model, which was trained on a custom dataset made of samples from movie production recordings from the Pro Sound Effects Library database (Pro Sound Effects, 2021). We use the model given as a baseline for the FUSS dataset (Wisdom et al., 2021), which is able to separate mixtures with a variable number of sources, ignoring the outputs that have a power below a given threshold.

3.3 Testing

The procedure we use for testing is as follows:

1. The target is created as a combination of four randomly chosen sources, which are offset to begin playing at different times.
2. The target, *without any separation performed*, is orchestrated using the entire orchestra. This creates what we call the **full target orchestration**.
3. The full orchestra is randomly split into four equal sized subsets, each containing 7 to 8 instruments.
4. For each separation method:
    1. The separation method is applied to the target, splitting the target into four sub-targets.
    2. The separation is evaluated using the cosine distance.
    3. The four sub-targets are separately orchestrated, each using a different subset of the orchestra.
    4. The four orchestrations are combined to play simultaneously, creating the **separated orchestration**.
    5. The distance between the target and the separated orchestration is calculated.

The orchestrations performed are static orchestrations, meaning the orchestration does not change over time; a single onset of notes is created for each sub-target, no matter if the target itself has multiple onsets. The OrchideaSOL database of orchestral samples is used with Orchidea to create the orchestrations (Cella et al., 2020b). This database contains recordings of extended playing techniques, which better fits the often noisy or inharmonic nature of our targets. The full orchestra, of which non-overlapping subsets were selected to orchestrate the sub-targets, contains the following instruments: 8 violins, 4 violas, 3 cellos, 1 bass, 2 oboes, 2 flutes, 2 clarinets in B♭, 2 bassoons, 2 trumpets, 2 trombones, and 2 French horns.

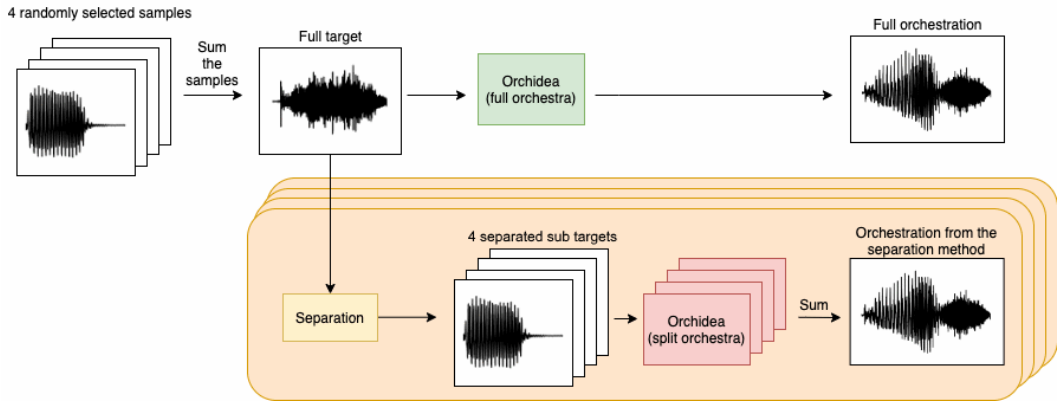

Figure 2: Diagram showing how the full target orchestration is a single orchestration with the full orchestra, and the separated orchestration is four orchestrations with four subsets of the orchestra.

3.4 Evaluation

We compare the effectiveness of the separation methods in two ways. First, we evaluate the separation accuracy by calculating the cosine distance between the target and the output of a separation method. Second, we compare how each separation method affects the quality of the orchestration by calculating the spectral distance between the target and the separated orchestration. We use these distance metrics as they are specific to our problem of computer-assisted orchestration (Abreu et al., 2016).

3.4.1 Separation evaluation

The source separation step of our experiments was evaluated independently from the orchestration process. To evaluate the source separation quality of an estimate-reference pair, we first frame the waveforms into non-overlapping segments of approximately 90ms. Then, for each frame, we compute the normalized magnitude spectrum and compare them using the cosine distance. Therefore, the distance between an estimated sub-target $\tilde{x}$ and a reference x is given by Eq. 1.

$$D(x, \tilde{x}) = \frac{1}{n} \sum_{i}^{n} (1 - \frac{\langle x_i, \tilde{x_i} \rangle}{\|x_i\| \cdot \|\tilde{x_i}\|}) \tag{1}$$

where x_i is the normalized magnitude spectrum of the *i*-th frame. To align the estimates from the source separation models with their corresponding reference sub-targets, we consider all references-estimations permutations, and we keep the best result.

3.4.2 Orchestration evaluation

Our goal is to evaluate the effect of source separation methods on orchestration. We determine the quality of an orchestration by calculating the spectral distance between the target and orchestration. The spectral distance metric is proposed in Cella (2020) as part of the cost function used in Orchidea during the optimization, and used in Cella et al. (2020a) to compare accuracies of orchestrations. The equation takes in the normalized magnitude spectrum of the target x and of the solution $\tilde{x}$. Then for each bin k of the spectrum, it calculates the absolute difference between the amplitudes. The differing values of λ_1 and λ_2 allow the metric to penalize the solution in different ways. For our purposes, we used $\lambda_1 = 0.5$ and $\lambda_2 = 10$, which penalizes a solution that overshoots the harmonic energy of the target.

$$d(x, \tilde{x}) = \lambda_1 \sum_{k} \delta_{k1}(x_k - \tilde{x}_k) + \lambda_2 \sum_{k} \delta_{k2}|x_k - \tilde{x}_k| \tag{2}$$

where $\delta_{k1} = 1$ if $x_k \geq \tilde{x}_k, 0$ otherwise; and $\delta_{k2} = 1$ if $x_k < \tilde{x}_k, 0$ otherwise.

We found that sometimes one or more of the outputs of a separation method would be silence. This happens when the method cannot distinguish between two or more sources and incorrectly identifies

Table 1: Results for separation and orchestration evaluation averaged across 300 targets. Separation evaluations uses the cosine distance defined in Eqn. 1. Orchestration evaluation uses the spectral distance defined in Eqn. 2. "Full target" corresponds to the orchestration of the target with no separation performed (see Sec. 3.3). Lower values are better for both metrics.

Separation method	Separation evaluation	Orchestration evaluation
TDCN++	0.924	8.59
Conv-Tasnet	0.816	8.13
Open-Unmix	0.668	7.21
Demucs	0.655	6.90
NMF	**0.633**	**5.61**
Full target	-	9.24

them as the same source. Therefore, some sub-targets can have multiple sources and some sub-targets can be silence. However, Orchidea still attempts to orchestrate silence, so when a sub-target is identified as silence, the orchestration of that sub-target is not included in the separated orchestration. We identified silent sub-targets by calculating their root mean square (RMS) value. If the RMS was below a threshold of 0.05 then the sub-target orchestration was not included in the separated orchestration.

$$RMS(x) = \sqrt{\frac{1}{n}\sum_{i}^{n} x_i^2} \tag{3}$$

To determine whether source separation improves orchestration, we compare the distance between the target and the full target orchestration to the distance between the target and the separated orchestration. If the separated orchestration distance is lower, then performing separation improved the orchestration.

4 Results and discussion

The separation evaluation and orchestration evaluation results, averaged across 300 targets, are displayed in Tab. 1. The results show that performing source separation followed by a static orchestration of each individual sub-target improves the orchestration for all methods tested. Furthermore, when comparing the results between the separation evaluation and orchestration evaluation, we find that as the quality of the separation increases, so does the quality of the orchestration. In fact, the relative rankings between each method are exactly the same when evaluating separation and orchestration. NMF shows the best results, it has the lowest cosine distance for the separation evaluation and shows a 39% decrease in spectral distance compared to the full target orchestration.

A surprising result is the difference in accuracy for the two universal source separation methods: NMF and TDCN++. For both the separation evaluation and the orchestration evaluation, NMF performed the best out of all methods and TDCN++ performed the worst. We expect universal source separation to fit our problem better than music source separation, since our targets contain many non-musical sounds. However, our data suggests that the aspect of the separation method that affects orchestration is not whether it is universal or music source separation, but whether it is a supervised or unsupervised method.

The one unsupervised method we tested, NMF, outperformed all the supervised methods. This suggests that either unsupervised methods are better for our task, or that the data the supervised methods were trained on does not fit our problem well. It is important to note that for the supervised methods, we used pre-trained versions of the neural models. All of these models, except for TDCN++, were trained on musical data. However, we tested them on a wide range of targets, many of which were not strictly musical. This could explain why Demucs, Open-Unmix, and Conv-Tasnet performed worse than NMF. Similarly, one reason that TDCN++ may have performed worse than the other supervised methods is because it was trained on a different database than the other three supervised methods.

5 Conclusions and future work

In this paper, we addressed the potential of using source separation as a pre-processing step for computer-assisted orchestration. Based on the concept of dovetailing, our hypothesis is that separating a multi-layered sound and orchestrating the individual layers should result in an orchestration that better matches the timbre of the target. Our data confirms our hypothesis: all of the separation methods tested improve the resulting orchestration. Furthermore, there is a correlation between the effectiveness of the separation and the quality of the orchestration: as separation improves, so does the orchestration. This shows that our approach and metrics are consistent.

We also performed a qualitative evaluation of the orchestrations. An interesting result that we discovered from acoustic inspection is that the orchestrations generated after source separation tend to favor dovetailing. Since only a portion of the orchestra is assigned to each source and since the sources are inherently asynchronous with each other, the final orchestration is more fluid and appears more interesting from a musical standpoint.

More work is needed to identify why TDCN++, a universal source separation method, performed worse than the musical source separation models. Its performance is likely a result of the data it was trained on, which was not necessarily optimized for the sounds common to computer-assisted orchestration. A next step could be training this architecture ourselves on data that fits our problem better.

Another aspect that could be improved is the assignment of the orchestras. Currently, the subset of the orchestra that is assigned to each sub-target is randomly selected. However, it is possible that this random assignment could reduce the quality of the final orchestration. If, for example, one of the sub-targets contains mostly low pitch content, but the orchestra assigned to it contains mostly high-pitched instruments, the resulting orchestration will suffer. To solve this problem, a step could be added that jointly-optimizes the orchestras assigned to each sub-target, ultimately choosing an assignment that leads to the best orchestration of each sub-target.

As we stated in Section 1, we performed only static orchestration in our experiments in order to better observe the effects of source separation. However, dynamic orchestration is an important feature of computer-assisted orchestration and could greatly benefit from source separation. Future work could include applying source separation to dynamic targets and performing dynamic orchestration.

Finally, one or multiple of the separation methods we tested should be implemented in Orchidea in order to improve the orchestrations Orchidea is capable of creating.

References

Abreu, J., Caetano, M., and Penha, R. (2016). Computer-aided musical orchestration using an artificial immune system. In Johnson, C., Ciesielski, V., Correia, J., and Machado, P., editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 1–16, Cham. Springer International Publishing.

Adler, S. (2016). *The study of orchestration*. W. W. Norton and Company, New York, fourth edition.

Ariza, C. (2005). Navigating the landscape of computer aided algorithmic composition systems: a definition, seven descriptors, and a lexicon of systems and research. In *ICMC*.

BBC (2021). BBC sound effects archive. `https://sound-effects.bbcrewind.co.uk/`. Accessed Feb. 2021.

Cella, C.-E. (2020). Orchidea: a comprehensive framework for target-based assisted orchestration. *submitted to Journal of New Music Research, under review*.

Cella, C. E., Dzwonczyk, L., Saldarriaga-Fuertes, A., Liu, H., and Crayencour, H.-C. (2020a). A study on neural models for target-based computer-assisted musical orchestration. In *2020 Joint Conference on AI Music Creativity (CSMC + MuMe)*, Proceedings of the 2020 Joint Conference on AI Music Creativity, Stockholm, Sweden.

Cella, C.-E. and Esling, P. (2018). Open-source modular toolbox for computer-aided orchestration. In *Timbre conference*, Montreal, Canada.

Cella, C. E., Ghisi, D., Lostanlen, V., Lévy, F., Fineberg, J., and Maresz, Y. (2020b). OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration. In *International Computer Music Conference*.

Chordia, P. and Rae, A. (2009). Using source separation to improve tempo detection. In *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 183–188.

Cichocki, A. and Phan, A.-H. (2009). Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions*, 92-A:708–721.

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

Fernandez, J. and Vico, F. (2013). AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582.

Font, F., Roma, G., and Serra, X. (2013). Freesound technical demo. In *ACM International Conference on Multimedia (MM'13)*, pages 411–412, Barcelona, Spain. ACM.

Févotte, C. and Idier, J. (2011). Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, 23(9):2421–2456.

Gómez, J. S., Abeßer, J., and Cano, E. (2018). Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. In *ISMIR*, pages 577–584.

Kavalerov, I., Wisdom, S., Erdogan, H., Patton, B., Wilson, K. W., Roux, J. L., and Hershey, J. R. (2019). Universal sound separation. *CoRR*, abs/1905.03330.

Luo, Y. and Mesgarani, N. (2018). Tasnet: Surpassing ideal time-frequency masking for speech separation. *CoRR*, abs/1809.07454.

Maresz, Y. (2013). On computer-assisted orchestration. *Contemporary Music Review*, 32.

Pro Sound Effects (2021). Pro sound effects library. `https://www.prosoundeffects.com/`. Accessed: Feb. 2021.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The MUSDB18 corpus for music separation. `https://doi.org/10.5281/zenodo.1117372`.

Reed, J., Ueda, Y., Siniscalchi, S. M., Uchiyama, Y., Sagayama, S., and Lee, C.-H. (2009). Minimum classification error training to improve isolated chord recognition. In *ISMIR*, pages 609–614.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Stöter, F.-R., Liutkus, A., and Ito, N. (2018). The 2018 signal separation evaluation campaign. In *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK*, pages 293–305.

Stöter, F.-R., Uhlich, S., Liutkus, A., and Mitsufuji, Y. (2019). Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software*, 4(41):1667.

Trowitzsch, I., Taghia, J., Kashef, Y., and Obermayer, K. (2019). The NIGENS general sound events database. `https://doi.org/10.5281/zenodo.2535878`. Accessed: Feb. 2021.

Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. (2017). Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 261–265.

Wisdom, S., Erdogan, H., Ellis, D. P., Serizel, R., Turpault, N., Fonseca, E., Salamon, J., Seetharaman, P., and Hershey, J. R. (2021). What's all the fuss about free universal sound separation data? In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 186–190. IEEE.

Zapata, J. R. and Gómez, E. (2012). Improving beat tracking in the presence of highly predominant vocals using source separation techniques: Preliminary study. In *Proc. 9th Int. Symposium on Computer Music Modeling and Retrieval, London*, pages 583–590.