IMPROVING SOURCE SEPARATION BY EXPLICITLY MODELING DEPENDENCIES BETWEEN SOURCES

Ethan Manilow[1,2,*], *Curtis Hawthorne*[1], *Cheng-Zhi Anna Huang*[1], *Bryan Pardo*[2], *Jesse Engel*[1]

[1]Google Research, Brain Team [2]Northwestern University

ABSTRACT

We propose a new method for training a supervised source separation system that aims to learn the interdependent relationships between all combinations of sources in a mixture. Rather than independently estimating each source from a mix, we reframe the source separation problem as an Orderless Neural Autoregressive Density Estimator (NADE), and estimate each source from both the mix and a random subset of the other sources. We adapt a standard source separation architecture, Demucs, with additional inputs for each individual source, in addition to the input mixture. We randomly mask these input sources during training so that the network learns the conditional dependencies between the sources. By pairing this training method with a blocked Gibbs sampling procedure at inference time, we demonstrate that the network can iteratively improve its separation performance by conditioning a source estimate on its earlier source estimates. Experiments on two source separation datasets show that training a Demucs model with an Orderless NADE approach and using Gibbs sampling (up to 512 steps) at inference time strongly outperforms a Demucs baseline that uses a standard regression loss and direct (one step) estimation of sources.

***Index Terms*—** music source separation, orderless NADE, Gibbs sampling

1. INTRODUCTION

Within an auditory scene, musical sources are highly coordinated; musicians create sounds that are intended to overlap in time (i.e., rhythm) and frequency content (i.e., harmony). This coordination is fundamental to most music and helps distinguish it from other types of auditory sources such as speech and environmental sounds. However, this coordination makes isolating individual sounds difficult, as done in music source separation. The synchronization of harmonic content can lead to overlapping frequency partials; if different instruments produce different notes in a chord voicing, it might be hard to determine which frequency belongs to which source. For example, the root of a chord might be played by a bass guitar and the rest of the chord played by a piano. In this case, the higher partials of the bass might be hard to distinguish from those that came from the piano.

It stands to reason, therefore, that complete or partial knowledge about one source could be helpful when estimating another. This contextual knowledge might be essential in the case of the bass guitar and piano above: if all or some of the partials of the piano notes are known, a system might be able to make a better estimate of the bass. In probability theory, this process is termed "explaining away", where possible explanations are eliminated as knowledge is acquired. While explaining away might be especially important for musical signals, we note that our approach might be broadly useful even in unstructured audio scenes.

*Work done as a Google Student Researcher.

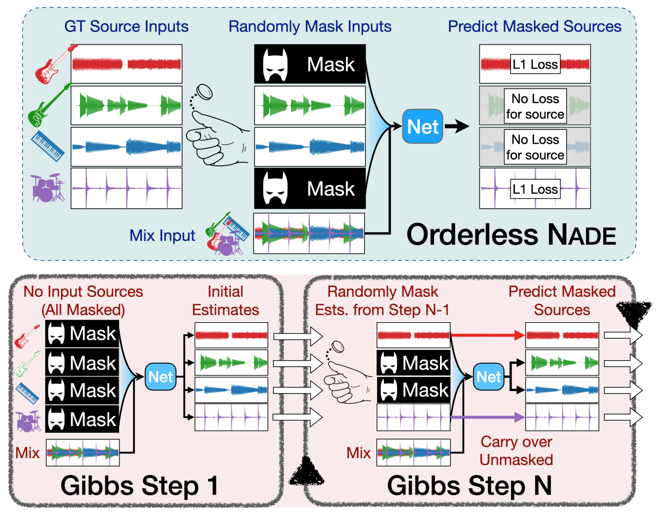

Fig. 1: The proposed Orderless NADE training (top) and blocked Gibbs sampling procedure (bottom). For Gibbs Step N, un-masked source inputs are the model's source estimates from Step N-1.

Despite this, deep learning-based music source separation research has largely ignored complementary context between musical sources, opting instead to model each source independently conditioned only on the mix. More often than not, this means a separation system is designed to separate a single source given only the mixture as input, with no consideration given to explaining away information provided by other sources. In this paper, we introduce a new means of training and sampling from a source separation system such that it is able to leverage contextual information provided by other sources.

We propose alterations to the training and inference procedures of existing neural networks to account for contextual information between different sources in a musical scene. Specifically, we propose to model the dependencies between different sources within a mixture by autoregressively factorizing the conditional probability distribution, $p(\mathbf{s}_1, \ldots, \mathbf{s}_I | \mathbf{m})$, of a mixture $\mathbf{m}$ and sources, $\mathbf{s}_1, \ldots, \mathbf{s}_I$, in such a way that a model learns useful conditional dependencies between all combinations of sources within a mix. To accomplish this, we alter a separation network with additional waveform inputs used for conditioning the net with contextual source data. During training, we provide ground truth source data to a random subset of these conditioning inputs (i.e., teacher forcing) and task the network with estimating the sources not given as input following an Orderless Neural Autoregressive Density Estimator (NADE) [1, 2] formulation (see Section 3). At inference time, we run a blocked Gibbs sampling procedure by first making a set of initial source estimates, and then iteratively feeding those estimates back into the network to refine and improve them.

This training and inference procedure is agnostic to the choice of model architecture, therefore we experimentally verify our method using an existing music separation network. We show that the proposed training and sampling procedure substantially increases separation performance on two source separation datasets. We show the effect of Gibbs sampling and, in many cases, the performance of source estimates monotonically increase no matter how many Gibbs sampling steps we try. In other cases, source estimates improve over baseline training after one inference step (i.e., no Gibbs sampling), indicating that our Orderless NADE training is beneficial by itself.

2. PRIOR WORK

In the typical source separation formulation, the goal is to predict some source given an input mixture. In other words, to produce a source estimate, a model is *only* conditioned on the mix, making each estimate produced independently of any other source in the mix.

This formulation leads many systems to treat each source independently, leading some top performing systems to independently train a new network for each source (i.e, "one-vs-all") [3, 4, 5], effectively treating each network as a source denoiser. Other systems output multiple sources at once (i.e., "multi-source") [6, 7], potentially enabling a net to implicitly learn relationships between the sources, but ultimately still only conditioning the net on the mix. In this work, we explicitly condition a net on other sources in the mix.

A number of recent music separation systems condition on auxiliary information as a means of assisting the separation process. For instance, some systems condition on the linguistic content of singing [8], the musical score [9], class ID of a source [10, 11, 12, 13], or an audio query input [14, 15] as a means of steering the network to separate different sources. In this work, we condition on other sources within the same mixture as a means of learning dependencies between different sources in a mix, without auxiliary information. In Section 4 we describe how this enables a model to increase its separation performance by bootstrapping from its initial estimates.

Some recent work has taken a more generative modeling perspective on source separation, similar to our proposed autoregressive approach. For instance, Jayaram & Thickstun [16] propose a fast sampling method for a WaveNet [17] such that it can be used in a one-vs-all separation setup. WaveNet is a causal, autoregressive generative model for raw audio that produces a waveform, x, such that the output sample x_t at time t is conditioned past predicted samples, $p_\theta(x_t|x_1, \ldots, x_{t-1})$. This setup is autoregressive *in time* (within a source) but not *between* sources. In contrast, our work does not model dependencies in time (within a source), but does model dependencies between sources. Finally, Demucs [18] is a source separation network inspired by generative modeling, whereby the network forgoes a masking step in favor of directly estimating the waveforms of multiple sources. In this paper, we extend Demucs v2 to model dependencies between sources.

Our work draws inspiration from COCONET [19], which is a generative model of musical scores and uses contextual note information to infill new musical voices. We expand the scope of their Orderless NADE [1, 2] formulation to the continuous domain of audio signals, by modeling sources in a source separation context. Our method can alternatively be viewed as similar to a masked language model (MLM) setup, used for training models like BERT [20] and T5 [21]. These models mask words to learn dependencies within language corpora, whereas we mask sources to learn dependencies within mixes. Our work can also be seen as fitting under the umbrella of order agnostic diffusion models [22, 23], the main difference being that in our case each source is considered a discrete maskable variable, albeit with many continuous dimensions.

3. ORDERLESS NADE TRAINING

Source separation is the task of conditionally predicting a set of sources $\mathbf{s}_1, \ldots, \mathbf{s}_I$, given a mixture, $\mathbf{m}$, like $p_\theta(\mathbf{s}_1, \ldots, \mathbf{s}_I|\mathbf{m})$, for some model p_θ with parameters θ. Typically, this conditional distribution is independently factorized (Eq. 1), resulting in an independent training objective (Eq. 2):

$$p_\theta(\mathbf{s}_1, \ldots, \mathbf{s}_I|\mathbf{m}) = \prod_{i=1}^{I} p_\theta(\mathbf{s}_i|\mathbf{m}) \tag{1}$$

$$\mathcal{L}_{indep.} = -\sum_{i=1}^{I} \log p_\theta(\mathbf{s}_i|\mathbf{m}). \tag{2}$$

For example, the L1 loss used by Demucs corresponds to estimating the negative log likelihood of the waveform under a Laplacian distribution. Many systems even train one network per source, solidifying the independent factorization as a hard prior into the system's design. We note that the dimensionality of the sources $\mathbf{s}_1, \ldots, \mathbf{s}_I$ and mixture $\mathbf{m}$ are intentionally omitted because this formulation is agnostic to their domain (i.e., waveform, spectrogram, etc).

However, in this work we use a different way to factorize this conditional dependency, aiming to explicitly capture coordination between sources. As such, we want to condition each source on all possible combinations of other sources in the mix. To accomplish this, we propose a new way of training source separation systems.

Just as before, we want to conditionally predict a set of sources given a mix, $p_\theta(\mathbf{s}_1, \ldots, \mathbf{s}_I|\mathbf{m})$. Specifically, for the set of sources $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_I\}$, at each training iteration we randomly draw a subset of the sources, $D \subseteq S$, and then condition on this subset to predict sources in the complement set, $\neg D$. With $\mathbf{D}$ denoting the set of all random subsets of S, this enables us to factorize the conditional probability distribution like

$$p_\theta(\mathbf{s}_1, \ldots, \mathbf{s}_I|\mathbf{m}) = \prod_{\mathbf{D}} p_\theta(\mathbf{s}_{\neg D}|\mathbf{s}_D, \mathbf{m}). \tag{3}$$

As it turns out, this is an Orderless Neural Autoregressive Density Estimator (NADE) [1, 2], where we autoregressively predict sources in $\neg D$ conditioned on sources in D. As such, the loss is given by

$$\mathcal{L}_{\text{O-NADE}} = \mathop{\mathbb{E}}_{D \sim \mathbf{D}} - \sum_{i \in \neg D} \log p_\theta(\mathbf{s}_i|\mathbf{s}_D, \mathbf{m}), \tag{4}$$

where we apply a loss as in Eq. 1 to those sources in the compliment set $\neg D$. Note that this factorization requires that the model p_θ have different inputs than the factorization in Eq. 1: whereas before, the model only needed the mixture as input, in Eq. 3 the model now needs the mixture *and* the subset of sources in D as input. Because the sources in the complement set $\neg D$ are not used as input conditioning, we say that these sources are *masked*.

By masking a subset of its source input, the network must learn to use the unmasked part of its input as a conditioning signal for predicting the masked sections. With the proposed training setup the network learns the context surrounding the masked segments: it must use the unmasked context to predict the masked segments, therefore learning dependencies between sources. During training, loss is only computed for the source estimates that are masked, and the additional source inputs are teacher forced for simplicity, i.e., ground truth source data is used as input. Our Orderless NADE training procedure is shown at the top of Figure 1.

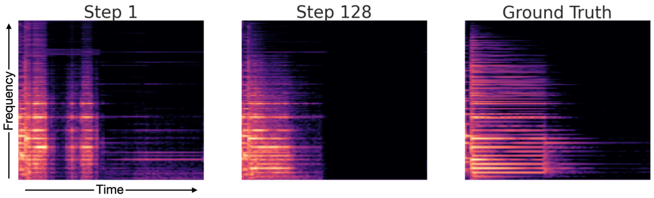

Fig. 2: Spectrograms showing a piano estimate after the 1st Gibbs step (left) and the 128th Gibbs step (middle). A spectrogram showing the ground truth signal is on the right. The Gibbs sampling iteratively improves the source estimation.

4. BLOCKED GIBBS SAMPLING

The addition of the extra conditioning information used by Orderless NADE training requires an inference procedure to match. The goal of the inference procedure is to improve initial estimates as sampling goes on, leveraging earlier estimates as conditioning information for the next step.

To that end, we use a blocked Gibbs sampling procedure at inference time. Standard Gibbs sampling is a Markov chain Monte-Carlo (MCMC) algorithm for estimating a multivariate probability distribution, whereby at each step one dimension is held out and the remaining dimensions are used as conditioning. Blocked Gibbs sampling holds out a set of dimensions instead of just one, as in standard Gibbs sampling. Following Eq. 3, the goal of the proposed system is to model the conditional distribution of all sets of randomly masked sources, $\mathbf{s}_{\neg D}$, given a mix, $\mathbf{m}$, and a corresponding set of complementary unmasked sources, $\mathbf{s}_D$, like $p_\theta(\mathbf{s}_{\neg D}|\mathbf{s}_D, \mathbf{m})$. Therefore we consider each complete source waveform a discrete maskable variable (albeit containing many continuous dimensions) during the sampling process. During inference, the mixture is always provided, unmasked, throughout the sampling process. Source estimates from previous steps are used as inputs for the next step, enabling the model to bootstrap its initial estimates for better results.

During sampling, an annealing schedule is used such that at earlier steps input sources are independently masked with a higher probability than later steps. At each step, sources are randomly masked with some probability, ρ, that decreases linearly over the sampling process according to the annealing schedule defined by Yao et. al. [2]. At step 0, $\rho = 1.0$ and at final step N, $\rho = 0.0$. We set $\alpha = 0.95$ and test various values for the number of steps, N.

5. EXPERIMENTAL VALIDATION

We conducted a set of experiments to validate the proposed training and testing setup. In our experiments, we tested the effect of training a separation network using our Orderless NADE and blocked Gibbs sampling procedures.

In our main experiments, we varied the number of Gibbs sampling steps at inference time. These experiments were designed to give us an understanding of the dynamics of the proposed blocked Gibbs sampling procedure for different lengths of sampling. We measured the source separation performance after 1, 4, 16, 64, 128, 256, and 512 Gibbs steps. We trained one network for each of the two datasets that we test: MUSDB18 [24] and Slakh2100 [25].

The first dataset we examined was MUSDB18 [24]. MUSDB18 consists of 150 mixtures and corresponding source from real recording sessions featuring live musicians. 100 of these are for training, from which we reserved 10 as a validation set. The remaining 50 were used for evaluation. The second dataset we focused on is Slakh2100 [25]. Slakh2100 contains 2,100 mixtures with corresponding source data that were synthesized using professional-grade sample-based synthesis engines. We train on 1289, use 270 for validation and evaluate on the 151 mixes in the test set.

For both datasets, we downsampled the audio to 16kHz. We segmented the audio into 4 second windows with a 2 second hop. We only kept windows where 2 or more sources are active. For MUSDB18, we defined a source as active if it has an RMS amplitude above –60dB, and all of the audio was converted to mono. We augmented MUSDB18 by applying pitch shifting and time stretching. For Slakh2100, we defined an active source as having more than 5 note onsets in the corresponding MIDI. We did not use any augmentation with Slakh2100, and the audio is mono. We used the bass, drums, guitar, and piano sources from Slakh2100.

We implemented our own Demucs v2 [18] waveform-to-waveform architecture as our source separation system. Demucs follows a U-Net pattern, with 6 encoder and decoder layers that have skip connections to between corresponding encoder and decoder layers. At its bottleneck, Demucs has an bidirectional LSTM with the same dimensionality as the last encoder layer. Demucs outputs multiple sources as waveform data. We refer the reader to the Demucs paper for full details [18]. In our implementation, we omitted many of the tricks that Demucs proposed to boost its performance (e.g., source remixing, weight rescaling, oversampling the audio). Demucs proposed a “shift trick”, which averaged a fixed number of forward passes at random time offsets for inference, similar to our Gibbs sampling procedure. In our implementation, we found that the shift trick was detrimental to performance, and thus omitted it in our experiments. We trained using L1 loss on the waveform using Adam with a learning rate of 3e-4 and batch size of 64 on 16 TPUv2 cores. We trained the Slakh2100 model for 100k steps and the MUSDB18 model for 85k steps.

Our Demucs was altered so that it had 8 additional input channels alongside the mixture. The first 4 channels were for injecting source estimates during training or sampling, and the final 4 channels (same dimensionality as the sources) provided as a sentinel flag to alert the network if any of the 4 input sources are masked. We compare this system to a baseline system trained in the typical manner, using only the mix as input, without Orderless NADE training or Gibbs sampling.

We conducted an additional experiment to test if models using the proposed training technique can effectively use the extra conditioning information. To do this we injected the ground truth source data for one source and measured the separation quality of the other sources after one Gibbs step. Because the model has perfect information for a given source, this serves as an upper bound on performance after one step. We compared this to the case where all of the sources are masked at the first step. We performed this on the Slakh2100 dataset using the same model as the above Slakh2100 experiment.

We measured the performance of all of these systems using the improvement in scale-invariant source-to-distortion ratio (SI-SDRi) [26] over the unprocessed mixture.

6. RESULTS AND DISCUSSION

The results of our main experiment are shown in Figure 3, which show plots with SI-SDR improvement as a function of number of Gibbs steps for each source in both datasets. The baseline system is shown as a dotted horizontal line for all sources. The top row shows results for the model trained on Slakh2100 sources and bottom row shows results for the model trained on MUSDB18 sources. The first

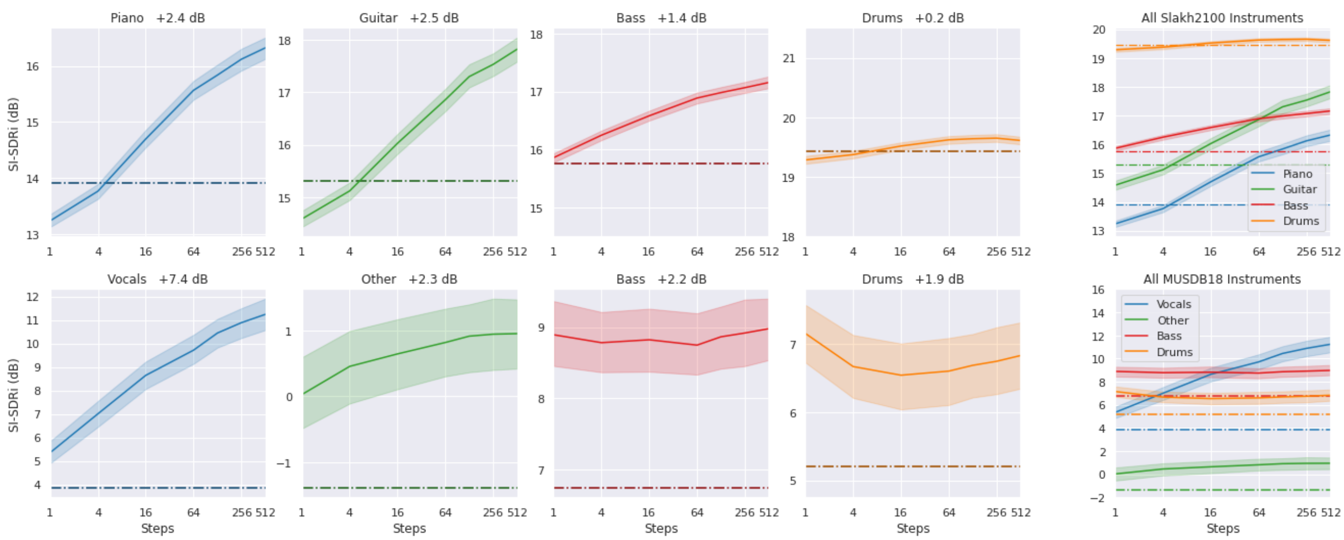

Fig. 3: Separation performance, in terms of SI-SDR improvement (dB), for the Slakh2100 (top row) and MUSDB18 (bottom row) models after different numbers of Gibbs steps. Higher is better. Solid lines are means of our proposed system, shaded areas represent a 95% CI, and dotted lines are mean of the baseline system. Note that the x-axis is log scaled. For many cases, the network is able to leverage earlier estimates to better performance. Furthermore, the sources that did not improve over time show large improvements over the baseline after just 1 step, indicating that Orderless NADE training was beneficial by itself.

Injected GT Source	Estimated Source			
	Piano	Guitar	Bass	Drums
Piano	–	+7.1	+4.7	+2.8
Guitar	+6.2	–	+3.4	+1.9
Bass	+4.1	+3.9	–	+2.4
Drums	+1.8	+2.1	+2.3	–

Table 1: Increase over regular Gibbs sampling when injecting ground truth sources for the first Gibbs sampling step, on the Slakh test set, in terms of mean SI-SDRi (dB). Sources in the same pitch register, like piano and guitar, see the biggest increase.

thing that we note is that the proposed training and sampling procedure always beats the baseline model after the first 16 Gibbs steps for both datasets. In fact, across the two datasets we test, 6 of the 8 total sources show that more Gibbs steps produces better separation results. However, our Gibbs sampling is not parallelized, so inference time scales linearly with the the number of Gibbs steps; after roughly 64 steps the sampling process has diminishing returns. Additionally, for MUSDB18, all sources show increased separation performance after the first Gibbs step–roughly +1–2 dB–indicating that the Orderless NADE training by itself is beneficial.

The most striking increase we see comes from the vocals source in the MUSDB18 dataset, which increases by +7.4 dB after 512 Gibbs steps. The largest increases we see in the Slakh2100 dataset come from the piano and guitar sources, that improve by over +2.0 dB after 256 steps. These two sources can occupy the same pitch register in a mix, so we expect that information about one source will help when estimating the other. Drums in the Slakh2100 dataset is the only source that does not see significant improvement. We hypothesize that this is because the performance of the baseline Demucs model is already so high (it has the highest mean SI-SDRi across all sources in both datasets), so there is less room to improve. Figure 2 is an illustrative example showing spectrograms of a piano estimate after 1 Gibbs step and 128 Gibbs steps.

The results in Figure 3 indicate that a standard source separation architecture is able to bootstrap its way to better separation performance–even after training concludes–when the network is adapted to use our proposed training and inference procedure. This is an indication that network is able to learn and leverage dependencies between musical sources within a mixture.

In Table 1, we show the results of our experiment to determine how effective the training procedure is at producing a network that learns the conditional dependencies in a mix. Results after one Gibbs step show that injecting ground truth for any source improves the output of all other sources. This indicates that a model trained with our procedure has learned how to effectively use the extra input information. The largest boost comes from estimating Guitar when ground truth Piano data is injected, and vice versa. These sources often have similar frequency content, meaning that any knowledge about one will frequently help the estimate of the other.

7. CONCLUSION

In this work, our goal was to model the interdependent relationships between different sources in musical mixtures. This is desirable because most musical mixes contain sources that are intentionally highly coordinated. To this end, we applied an Orderless NADE training procedure and a blocked Gibbs sampling procedure for music source separation. An Orderless NADE setup enables us to conditionally model the relationship between any two sources in a musical mixture. The blocked Gibbs sampling procedure allows the network to boost the separation performance by bootstrapping from previous estimates. These training and sampling procedures are agnostic to network architectures, so we experimentally verified them using Demucs v2 on two source separation datasets. Our proposed training and sampling procedures increased performance for nearly all of the sources we tested. These results show the power of using even partial conditional information to analyze a musical scene. We are excited to see how other ideas from generative modeling can positively influence research in understanding musical scenes.

8. REFERENCES

[1] Benigno Uria, Iain Murray, and Hugo Larochelle, "A deep and tractable density estimator," in *Proceedings of the 31st International Conference on Machine Learning*. 2014, vol. 32 of *Proceedings of Machine Learning Research*, pp. 467–475, PMLR.

[2] Li Yao, Sherjil Ozair, Kyunghyun Cho, and Yoshua Bengio, "On the equivalence between deep NADE and generative stochastic networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 322–336.

[3] Stylianos Ioannis Mimilakis, Konstantinos Drossos, Joao F Santos, Gerald Schuller, Tuomas Virtanen, and Yoshua Bengio, "Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 721–725.

[4] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, pp. 1667, 2019.

[5] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, pp. 2154, 2020.

[6] Ethan Manilow, Prem Seetharaman, and Bryan Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 771–775.

[7] Ryosuke Sawata, Stefan Uhlich, Shusuke Takahashi, and Yuki Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 51–55.

[8] Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez, "Content based singing voice extraction from a musical mixture," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 781–785.

[9] Marius Miron, Jordi Janer Mestres, and Emilia Gómez Gutiérrez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *International Society for Music Information Retrieval (ISMIR) Conference*.

[10] Gabriel Meseguer-Brocal and Geoffroy Peeters, "Conditioned-u-net: Introducing a control mechanism in the u-net for multiple source separations," in *International Society for Music Information Retrieval (ISMIR) Conference*, 2019.

[11] Olga Slizovskaia, Leo Kim, Gloria Haro, and Emilia Gomez, "End-to-end sound source separation conditioned on instrument labels," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 306–310.

[12] Prem Seetharaman, Gordon Wichern, Shrikant Venkataramani, and Jonathan Le Roux, "Class-conditional embeddings for music source separation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 301–305.

[13] David Samuel, Aditya Ganeshan, and Jason Naradowsky, "Meta-learning extractors for music source separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 816–820.

[14] Jie Hwan Lee, Hyeong-Seok Choi, and Kyogu Lee, "Audio query-based music source separation," *arXiv preprint arXiv:1908.06593*, 2019.

[15] Ethan Manilow, Gordon Wichern, and Jonathan Le Roux, "Hierarchical musical instrument separation," in *International Society for Music Information Retrieval (ISMIR) Conference*, 2020, pp. 376–383.

[16] Vivek Jayaram and John Thickstun, "Parallel and flexible sampling from autoregressive models via langevin dynamics," 2021, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, pp. 4807–4818, PMLR.

[17] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[18] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[19] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck, "Counterpoint by convolution," *International Society for Music Information Retrieval (ISMIR) Conference*, 2019.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[22] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg, "Structured denoising diffusion models in discrete state-spaces," 2021.

[23] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans, "Autoregressive diffusion models," 2021.

[24] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[25] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 45–49.

[26] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "Sdr–half-baked or well done?," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.