SOURCE SEPARATION BY STEERING PRETRAINED MUSIC MODELS

Ethan Manilow[1], *Patrick O'Reilly*[1], *Prem Seetharaman*[2], *Bryan Pardo*[1]

[1]Northwestern University [2]Descript, Inc.

ABSTRACT

We showcase a method that repurposes deep models trained for music generation and music tagging for audio source separation, without any retraining. An audio generation model is conditioned on an input mixture, producing a latent encoding of the audio used to generate audio. This generated audio is fed to a pretrained music tagger that creates source labels. The cross-entropy loss between the tag distribution for the generated audio and a predefined distribution for an isolated source is used to guide gradient *ascent* in the (unchanging) latent space of the generative model. This system does *not* update the weights of the generative model *or* the tagger, and only relies on moving through the generative model's latent space to produce separated sources. We use OpenAI's JUKEBOX as the pretrained generative model, and we couple it with four kinds of pretrained music taggers (two architectures and two tagging datasets). Experimental results on two source separation datasets, show this approach can produce separation estimates for a wider variety of sources than any tested system. This work points to the vast and heretofore untapped potential of large pretrained music models for audio-to-audio tasks like source separation.

***Index Terms*—** music source separation, generative music models, automatic music tagging, gradient ascent

1. INTRODUCTION

The research area of Music Information Retrieval (MIR) is constrained by a lack of labeled data sets, which limits our ability to train robust systems and evaluate them well. Specifically, the task of musical source separation has been hindered by a dearth of well-labeled data [1]. This leads to severe shortcoming in terms of the range of instrument source classes that current systems can separate. Many systems, in fact, only separate the four classes (voice, bass, drums and "other") in the widely-used MUSDB18 [2] dataset, making them unsuitable for separating most musical instruments.

Simultaneously, the recent availability of large pretrained models has revolutionized generative and discriminative tasks in the domains of computer vision and natural language processing. The combination of VQGAN [3] and CLIP [4] has captured the attention of many artists, who have been captivated by the system's ability to use natural language to create generative art. Similarly, researchers have shown how to steer large pretrained language models for downstream discriminative tasks either using transfer learning [5] or so-called few-shot "prompt engineering" [6]. Recent work has taken this ethos to the MIR domain, leveraging the representations learned by the large training regime of a generative music model for downstream MIR tasks, like key detection and music tagging [7].

In this work, we further this ethos by exploring how large, pretrained music models can be used for musical source separation, leveraging the vast amounts of unlabeled or weakly labeled data that these models see during training. We combine the VQ-VAE from OpenAI's JUKEBOX, a generative model of musical audio, with a music tagger. We task JUKEBOX with producing audio that matches a predefined set of tags that correspond with the musical source we wish to separate. To do this, we perform gradient ascent in the embedding space of the VQ-VAE and use the decoded audio as a mask on the input mixture. We demonstrate experimentally that this setup is able to separate a wider variety of sources than previous purpose-built separation systems consider, all without updating the weights of JUKEBOX or the tagger. We provide additional demos and runnable code on our demo site.[1]

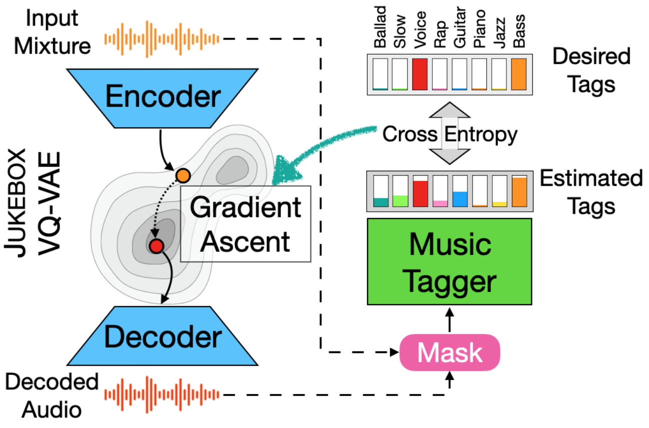

Fig. 1. Our system performs gradient ascent in the JUKEBOX VQ-VAE embedding space such that when the audio is input into a music tagger it matches a predefined set of tags. The weights of VQ-VAE and the Music Tagger are frozen. With this setup we can perform source separation.

2. PRIOR WORK

Recently, many source separation researchers have focused on methods that produce high-quality results on the datasets for which there is sufficient ground truth source data. For instance, the website Papers with Code shows a steady increase in the best performing separation systems on the MUSDB18 [2] dataset over the past few years.[2] Similarly, the recent Music Demixing Challenge [8] determined the best performing system on a test set that had the same source definitions as MUSDB18. As a result, the community has produced a large number of deep learning-based *supervised* separation systems that are purpose-built to separate sources as defined by MUSDB18.

[1] `https://ethman.github.io/tagbox`
[2] `https://bit.ly/pwc_musdb18`

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

However, the source definitions in MUSDB18 are limiting, [1] including isolated source data for only Vocals, Bass, Drums, and a catchall "Other" source for all other source types. Furthermore, MUSDB18 is relatively small, totalling 150 songs, which leads the authors of many systems [8, 9, 10] to collect additional data and lean heavily on augmentation.

Prior to the deep learning era, one of the most popular algorithms was Non-negative Matrix Factorization (NMF) [11]. While NMF is theoretically flexible enough to separate any source, it often required hand-designed algorithms to determine how to cluster spectral templates into coherent sources. Musical priors, such as repetition [12] or harmonicity vs. percussiveness [13], have also been used to create separation algorithms, however such algorithms are limited to separating sources that match the prior (e.g., a backing band) vs those that do not (e.g., a singing voice), and have been surpassed by deep learning-based methods. Although some recent neural network-based systems leverage the built-in priors of older algorithms for training [14, 15], our method does not rely on hand-designed priors, instead using the biases learned by generative music models and music taggers.

Similar to this paper is work by Jayaram and Thickstun [16], in which they propose a fast way to sample from autoregressive audio models, leading them to leverage the priors learned by a source-specific generative model to effectively denoise a mixture signal. To separate a new source, their work requires access to a large corpora of single-source audio to train a source-specific generative model. In contrast, our work separates new sources by simply changing the set of tag labels corresponding to a desired source.

Previous works have explored using additional networks for separation instead of directly optimizing a separation net on ground truth sources. For instance, the work of Pishdadian et. al. [17] is most similar to ours; they explore using a pretrained sound event detection (SED) system and the goal of the separator network is to maximize estimated SED labels during training. Similarly, Hung et. al [18] use a pretrained transcription network to train a separator. Our work differs from Pishdadian et. al. and Huang et. al. in that we do not *not* train any networks, instead we repurpose off-the-shelf networks that have *never* been trained for source separation.

3. BACKGROUND

3.1. OpenAI's JUKEBOX

OpenAI released JUKEBOX [19], a generative audio model that creates music. JUKEBOX is composed of two components: a hierarchical VQ-VAE [20] that learns to turn raw waveforms into discrete codes (called "tokens") and back, and a language model that learns how to generate new tokens which can be passed through the decoder to create musical audio. In this work, we are interested in the VQ-VAE, specifically.

JUKEBOX's VQ-VAE is a three-level hierarchical VQ-VAE that generates discrete tokens at different sample rates, compressing the 44.1kHz input audio to tokens with sample rates of 5.51kHz, 1.37kHz, and 344Hz for each level, respectively. Each level has a codebook size of 2048 with each code having 64 dimensions. All levels are trained to reconstruct the input waveform and are optimized with a multi-scale spectral loss. The VQ-VAE also uses a codebook loss to ensure that non-discretized latent vectors are close to their nearest neighbor discretized token vectors and a commitment loss, which stabilizes the encoder. The VQ-VAE is trained on 1.2 million songs scraped from the web. We refer the reader to the JUKEBOX paper for further training details [19]. Because we are interested in producing the highest-quality separation results possible, we only focus on the "Bottom" level, which compresses the input audio to tokens at a sample rate of 5.51kHz.

Algorithm 1 TAGBOX Optimization

```
Input: $\mathbf{x}$ input mixture, $\mathbf{L}_{src}$ desired source tags.
Output: $\mathbf{s}_{out}$ source estimate.
$\mathbf{e} \leftarrow V_{\text{enc}}(\mathbf{x})$ Encode the input mixture.
$\mathbf{X} \leftarrow STFT(\mathbf{x})$
repeat
    $\mathbf{j} \leftarrow V_{\text{dec}}(\mathbf{e})$ Decode the embedding.
    $\mathbf{J} \leftarrow STFT(\mathbf{j})$
    $\bar{\mathbf{M}}_{\mathbf{j}} \leftarrow \frac{|\bar{\mathbf{J}}|}{\max(|\mathbf{J}|,|\mathbf{X}|)+\varepsilon}$ Build the mask.
    $\bar{\mathbf{S}} \leftarrow \bar{\mathbf{M}}_{\mathbf{j}} \odot \mathbf{X}$ Mask the mixture.
    $\mathbf{L}_{est} \leftarrow Tagger(iSTFT(\bar{\mathbf{S}}))$ Probability over tags.
    $\mathbf{e} \leftarrow \delta \nabla \mathcal{L}_{CE}(\mathbf{L}_{est}, \mathbf{L}_{src})$ Update the embedding.
until max steps
$\mathbf{s}_{out} \leftarrow \mathbf{x} - iSTFT(\bar{\mathbf{S}})$
```

3.2. Automatic Music Tagging

Music tagging is the task of labeling musical audio clips with semantic labels called "tags" [22, 23]. These tags are useful for music search and recommendation systems, enabling automatic labelling of large music corpora. The content that the tags represent can vary, sometimes indicating information about a song's genre, the song's mood or theme, or whether particular instruments are audible.

Music tagging systems are designed to predict a set of multi-hot, binary labels (i.e., tags) based on the acoustic contents of an input signal. Many recent works use convolutional neural networks at their core, varying the convolutional filter size and input representation of the audio [24]. Common datasets for music tagging are an order of magnitude larger than source separation datasets: MagnaTagATune (MTAT) [22] contains 25,877 30-second labeled audio clips ($\approx$21x more hours of audio than MUSDB18) and MTG-Jamendo (MTG) [23] contains 55,701 labeled audio clips with a minimum song length of 30 seconds ($\geq$ 46x more hours of audio than MUSDB18). We refer the reader to Won et. al. for an overview of recent advances in music tagging [24].

In this work, we use pretrained music taggers provided by Won et. al [24]. We examine using two pretrained music tagging systems, with each having a different input representation: FCN [25] with Mel spectrogram inputs, and HarmonicCNN [26], which inputs a variant of a constant-Q transform that has learnable filters. We also explore using taggers trained on different datasets, namely MagnaTagATune (MTAT) [22] and MTG-Jamendo (MTG) [23].

4. PROPOSED SYSTEM

At the heart of our proposed system are two components: a pretrained generative music model (i.e., JUKEBOX) and a pretrained music tagging model. Because our system combines music taggers and JUKEBOX, we call our system TAGBOX. The core idea is simple: when audio of an isolated source is input to a music tagger, only the instrument tags corresponding to that source should be active. Therefore, given an input mixture, we fix a set of predefined tags corresponding to the source(s) we want to separate, and iteratively optimize the audio output of a generative model until it matches those tags. Our approach is shown in Figure 1 and Algorithm 1.

Method	# Trainable Parameters	Neural Network?	MUSDB18 [2]			Slakh2100 [1]				
			Vocals	Bass	Drums	Bass	Drums	Guitar	Piano	Strings
Demucs v2 [9]	265M	✓	15.5	13.1	12.7	10.8	15.5	–	–	–
Open-Unmix [10]	35M	✓	15.0	11.9	11.6	9.8	14.4	–	–	–
Cerberus [21]	16M	✓	–	8.3	7.5	10.8	15.4	10.2	10.5	12.5
HPSS [13]	0		–	–	−0.1	–	0.3	–	–	–
REPET-SIM [12]	0		7.8	–	–	–	–	–	–	–
TAGBOX (Ours)	1	✓	7.4	7.1	5.9	6.9	7.3	9.3	8.7	10.5

Table 1. Comparison of source separation systems in terms of mean SDR improvement (dB) over the unprocessed mixture. Grey cells indicate that the system is unable to separate that source type. TAGBOX is the only system that is able to separate *all* of the sources we test.

Given an input mixture waveform $\mathbf{x} \in \mathbb{R}^{1 \times t}$ of length t samples, a music instrument tagger returns a label vector $\mathbf{L}$, where the ith element of $\mathbf{L}$ gives the probability that instrument i is present in $\mathbf{x}$. We first create a target tag distribution $\mathbf{L}_{\text{src}}$ by setting the tags that correspond to the desired instrument sources to 1 (e.g., "guitar" or "drums") and all other tags to 0. $\mathbf{L}_{\text{src}}$ is an input parameter to the system that tells it which sources we want it to separate. We then use the encoder of an autoencoder, V_{enc}, to produce an initial embedding $V_{\text{enc}}(\mathbf{x}) = \mathbf{e} \in \mathbb{R}^{D \times \tilde{T}}$ from the input audio mixture $\mathbf{x}$, where $D \times \tilde{T}$ is the dimensionality of the embedding $\mathbf{e}$. Here, we use JUKEBOX's bottom-level VQ-VAE as the autoencoder, which has $D = 64$, and $\tilde{T}$ varies depending on the sample rates of the input audio (44.1 kHz) and tokens (5.51 kHz), and number of samples in the input audio. This embedding, $\mathbf{e}$, can be decoded back into a waveform $V_{\text{dec}}(\mathbf{e}) = \mathbf{j} \in \mathbb{R}^{1 \times t}$ by the decoder V_{dec}. Note that $V_{\text{dec}}(V_{\text{enc}}(\mathbf{x})) = \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}}$ is the autoencoder's reconstruction of $\mathbf{x}$, however in this work we optimize $\mathbf{e}$ such that the system can produce source estimates based on the desired source tags, $\mathbf{L}_{\text{src}}$.

Rather than pass the decoded audio $\mathbf{j}$ directly to the $Tagger()$, we use $\mathbf{j}$ to build a mask on the input mixture. Because $\mathbf{j}$ will be used to make a mask, the embedding $\mathbf{e}$ (via $V_{\text{dec}}(\mathbf{e}) = \mathbf{j}$) controls what information must be *removed* from the input mix to produce the desired source. To this end, we convert the input mix, $\mathbf{x}$, and JUKEBOX-decoded audio, $\mathbf{j}$, into spectrograms, $\mathbf{X} \in \mathbb{C}^{F \times T}$ and $\mathbf{J} \in \mathbb{C}^{F \times T}$, with F frequency bins and T time frames. We compute a real-valued mask, $\bar{\mathbf{M}}_{\mathbf{j}} \in [0.0, 1.0]^{F \times T}$ as follows:

$$\bar{\mathbf{M}}_{\mathbf{j}} = \frac{|\mathbf{J}|}{\max(|\mathbf{J}|, |\mathbf{X}|) + \varepsilon} \tag{1}$$

where $\max()$ is an element-wise max function between each time-frequency bin in a pair of spectrograms and a small epsilon, e.g ε=1e-8, prevents division by zero. This mask $\bar{\mathbf{M}}_{\mathbf{j}}$ is multiplied by the mixture spectrogram to get an estimate of the source spectrogram: $\bar{\mathbf{S}} = \bar{\mathbf{M}}_{\mathbf{j}} \odot \mathbf{X}$. Here, $\odot$ indicates element-wise multiplication. $\bar{\mathbf{S}}$ is then converted to a waveform of the source estimate $\bar{\mathbf{s}} \in \mathbb{R}^{1 \times t}$ using an inverse STFT.

This source estimate, $\bar{\mathbf{s}}$, is put into the music tagger to determine a set of tags from the source estimate, $Tagger(\bar{\mathbf{s}}) = \mathbf{L}_{\text{est}}$. We expect that as the source estimate, $\bar{\mathbf{s}}$, gets better, the source estimate tags, $\mathbf{L}_{\text{est}}$, will more closely match the predetermined tags representing our desired sources, $\mathbf{L}_{\text{src}}$. Therefore, we compute a binary cross-entropy loss $\mathcal{L}_{CE}(\mathbf{L}_{\text{est}}, \mathbf{L}_{\text{src}})$ between the source estimate tags, $\mathbf{L}_{\text{est}}$, and the desired instrument tags, $\mathbf{L}_{\text{est}}$. This loss is used to perform a gradient ascent step in the JUKEBOX embedding space, and the embedding is updated like $\mathbf{e} \leftarrow \delta \nabla \mathcal{L}_{CE}(\mathbf{L}_{\text{est}}, \mathbf{L}_{\text{src}})$ where δ governs the step size. We repeat this whole procedure for a predetermined number of optimization steps. Because the JUKEBOX-decoded audio, $\mathbf{j}$, determines what should be *removed* from the mix (via the mask $\bar{\mathbf{M}}_{\mathbf{j}}$), the final source estimate, $\mathbf{s}_{out}$, is the difference between the input mixture waveform $\mathbf{x}$ and the final $\bar{\mathbf{s}}$ produced by gradient ascent. The final source estimate is therefore $\mathbf{s}_{out} = \mathbf{x} - \bar{\mathbf{s}}$.

We note that *neither* the generative model *nor* the music tagger were trained for source separation and that no additional training or alteration of the weights of either model happens at any point. These models were, however, trained on datasets with a wider range of audio than is typical for purpose-built source separation systems.

Our system can produce separation results for more sources than any previous deep learning system that we are aware of. TAGBOX is limited only by the tags of the music tagging system, of which there are 12 distinct instrument tags in MTG-Jamendo (MTG). MagnaTagATune (MTAT) has 31 tags that could be interpreted as instrument tags, although some tags conceptually overlap (e.g., MTAT contains distinct tags for "Vocals", "Voice", "Male Vocals", etc). A corollary to this is that if there are no tags for a source, our system cannot separate it (e.g., MTAT has no "Bass" tag). Separating different source types does not require any changes to the system setup other than altering a set of predefined tags. Compare this to typical music separation networks like Open-Unmix [10] which would require training a whole model for each new source or Demucs [9] which would require altering the network architecture to add a new source output.

5. EXPERIMENTAL VALIDATION

We conduct two experiments to validate our system. The first and main experiment compares the proposed system to existing systems, taking special care to understand TAGBOX's ability to separate many types of sources. The second experiment shows how the choice of the pretrained, frozen Tagger model affects separation quality.

In our main experiment, we compare our system to existing systems on two established test sets for source separation, namely MUSDB18 [2] and Slakh2100 [1]. In this experiment we compare our proposed system against recent deep learning-based supervised separation systems as well separations based on musical priors. We compare a wide variety of source types across both datasets.

The first dataset we examine is MUSDB18. MUSDB18 contains 150 mixtures and corresponding sources from live recording sessions, 100 are reserved for training and the remaining 50 are used for testing. For this experiment, we exclude MUSDB18's "other" source because it could map to many possible tags. The supervised systems that we compare against, namely Open-Unmix [10] and Demucs [9], are trained using the MUSDB18 training set. Contrast this to HPSS [13] and REPET-SIM [12], which are run on the test set without any training. Our proposed system falls into this second

Tagger Settings		MUSDB18 [2]			Slakh [1]				
Dataset	Architecture	Vox	Bass	Drums	Bass	Drums	Guitar	Piano	Strings
MagnaTagATune	FCN	7.9	–	5.7	–	7.3	9.6	8.6	10.4
	HCNN	6.6	–	5.0	–	6.5	8.8	7.3	8.5
MTG-Jamendo	FCN	7.4	7.1	5.9	6.9	7.3	9.3	8.7	10.5
	HCNN	6.8	6.7	5.8	6.7	7.3	8.3	8.1	9.0

Table 2. Comparison of using different pretrained, frozen taggers for gradient ascent with TagBox in terms of mean SDR improvement (dB) over the unprocessed mixture. Note the MagnaTagATune taggers have no "Bass" tag.

camp; it also does not have a separation training phase.

The main experiment also uses the Slakh2100 [1] dataset. Slakh2100 contains 2100 mixtures with corresponding sources that were synthesized using professional-grade sample-based synthesis engines. We chose 50 songs from the test set to evaluate on. We chose songs that have source data for following five source types: bass, drums, guitar, piano, strings. We select mixes where all 5 sources are active, and we say a source is active if it has 100 or more note onsets throughout the entirety of the song, as determined by the corresponding MIDI data. We create mixes by instantaneously mixing together the sources and use these mixtures as input to the systems. With this setup we compare against Cerberus [21], which was trained to separate these five instruments, specifically.

For TagBox, we use a pretrained FCN [25] tagger trained on the MagnaTagATune (MTAT) [22] dataset. We run gradient ascent with a learning rate of 5.0 using the Adam optimizer for 10 steps (in the interest of brevity), and use a spectrogram with 1024 FFT bins for the mask. Additionally, we use the "foreground" from REPET-SIM as the vocals estimate, following prior work [12], and use the "percussion" output from HPSS as the drums estimate. We omit the other source outputs of these systems because they are ill-defined (e.g., HPSS's "harmonic" could be many possible sources).

In the second experiment, we compare four different configurations of our proposed system, varying the architecture and training data of the music tagger. We look at the FCN [25] and HarmonicCNN [26] architectures, trained either MagnaTagATune (MTAT) [22] or MTG-Jamendo [23]. We use the same learning rate and number of steps as the previous experiment. We evaluate the outcome of our experiments using the source-to-distortion ratio improvement (SDRi) over the unprocessed mixture [27], using the *museval* toolbox [28].

6. RESULTS AND DISCUSSION

Table 1 shows the results of our main experiment. In terms of SDRi, our system is better than or competitive with both of the hand-designed algorithms that we test against, HPSS and REPET-SIM. Additionally, while our system does not show as good of performance as the purpose-built supervised separation systems (i.e., Open-Unmix, Demucs, and Cerberus), it still shows a considerable SDRi boost for all sources that we test. Importantly, our system is able to boost performance over a wider array of source types than any other system we compare against.

The results from our second experiment are shown in Table 2. Of the two architectures we test, using FCN always produces better separation results. Interestingly, the opposite trend was observed when the taggers were evaluated for music tagging performance by Won et. al. [24]: HCNN was among the top performing systems and FCN was towards the bottom of the pack.

In many cases, TagBox can leave much to be desired perceptually; in most cases its separation performance is not up to the same level as the purpose-built separation systems we compare against. However, when listening to the output, there is no doubt that TagBox is able to separate the desired source, despite apparent artifacts. We have informally noticed a few tricks for better perceptual performance, like using multiple FFT sizes when making the masks (*à la* a multi-scale spectral loss) and doing gradient ascent for 100 steps. These perceptual tricks were however not reflected in the SDR evaluation numbers. Furthermore, because producing each output example requires its own gradient ascent, adding more steps increases the computation time linearly, which can be a costly process when run on an entire dataset. However, this might be tolerable for musicians needing a flexible source separation solution on a single song.

There are also a few other variants of the TagBox setup that can lead to fun and unexpected creative results. In the first case, we remove the masking step and allow TagBox to create audio freely, without the constraint of having to only remove information from the mix. With this setup, TagBox performs a kind of style transfer, mapping certain features of the audio to the desired tag. In one example, a mixture had a singer and we selected the "guitar" tag. TagBox made the resultant audio sound like a guitar was performing the melody. Additionally, another variant involves selecting non-instrument tags, like genre tags, and optimizing those.

What we find most impressive is that neither Jukebox nor the music taggers were trained for source separation. Furthermore, the weights of both networks do not change during the gradient ascent process; only the location of the audio in the Jukebox embedding space changes. The combination of Jukebox and the taggers have seen up to 1.25 million songs and combined these systems are able to leverage their shared priors about music and musical sources to isolate individual musical sources in a mixture. We believe that these priors could be leveraged in many ways to overcome the data scarcity problems endemic to many MIR tasks, as has already been investigated to great effect by Castellon et. al [7].

7. CONCLUSION

In this paper, we propose a method for source separation by combining pretrained models, called TagBox. We use pretrained music taggers to guide gradient ascent in the embedding space of OpenAI's Jukebox with the goal of maximizing a pre-defined tag corresponding to the source we want to separate. The output of Jukebox is used as a mask on the input audio before being sent to the tagger, which ensures that Jukebox does not generate new, unseen data that is not present in the input mixture. Importantly, neither the tagger nor Jukebox were trained for source separation and the weights of both models remain fixed during the gradient ascent process. Our results show that our system can separate a wider variety of source types than many recent purpose-built, supervised separation systems. We are excited by the promise that pretrained systems hold for source separation research.

8. REFERENCES

[1] 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.

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

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12873–12883.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[5] 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.

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[7] Rodrigo Castellon, Chris Donahue, and Percy Liang, "Codified audio language modeling learns useful representations for music information retrieval," *arXiv preprint arXiv:2107.05677*, 2021.

[8] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, and Fabian-Robert Stöter, "Music demixing challenge at ismir 2021," *arXiv preprint arXiv:2108.13559*, 2021.

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

[10] 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.

[11] Paris Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1-3, pp. 21–34, 1998.

[12] Zafar Rafii and Bryan Pardo, "Music/voice separation using the similarity matrix," in *In Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[13] Derry Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2010, vol. 13.

[14] Prem Seetharaman, Gordon Wichern, Jonathan Le Roux, and Bryan Pardo, "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 356–360.

[15] Prem Seetharaman, Gordon Wichern, Jonathan Le Roux, and Bryan Pardo, "Bootstrapping deep music separation from primitive auditory grouping principles," *arXiv preprint arXiv:1910.11133*, 2019.

[16] Vivek Jayaram and John Thickstun, "Parallel and flexible sampling from autoregressive models via langevin dynamics," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 4807–4818, PMLR.

[17] Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux, "Finding strength in weakness: Learning to separate sounds with weak supervision," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2386–2399, 2020.

[18] Yun-Ning Hung, Gordon Wichern, and Jonathan Le Roux, "Transcription is all you need: Learning to separate musical mixtures with score as supervision," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 46–50.

[19] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever, "Jukebox: A generative model for music," 2020.

[20] Ali Razavi, Aaron van den Oord, and Oriol Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in neural information processing systems*, 2019, pp. 14866–14876.

[21] 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.

[22] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie, "Evaluation of algorithms using games: The case of music tagging.," in *In Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 387–392.

[23] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The mtg-jamendo dataset for automatic music tagging," 2019.

[24] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra, "Evaluation of cnn-based automatic music tagging models," in *Proc. of 17th Sound and Music Computing*, 2020.

[25] Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," 2016.

[26] Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serrc, "Data-driven harmonic filters for audio representation learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 536–540.

[27] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito, "The 2018 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK*, 2018, pp. 293–305.