technically generative. However, prior work has shown that the unique mode-selecting behaviors of these adversarial loss models are highly effective for densely-conditioned generative modeling tasks such as vocoding [20,24,25] and speech denoising and enhancement [26–32]. Our goal is not an exact reconstruction of ground truth, but an output that is perceptually improved. This means a distribution of viable outputs exists and the task can be framed as one of generative modeling.

4. EXPERIMENTAL VALIDATION

We conducted experiments to understand whether MSG perceptually enhances the raw output of a set of music source separation models. The remainder of this section is devoted to outlining the details of our experiments.

4.1 Models

We trained MSG post-processors using the output of four existing source separation models as the input to MSG. We train on two waveform-based separators: Demucs v2 [7] and Wavenet [8]; and two spectrogram-based separators: Spleeter [17] and OpenUnmix [16]. To investigate whether MSG can learn to correct the artifacts of different separators, we created one enhancement model that is trained and evaluated on all four separators instead of creating separator-specific models. We evaluated the MSG post-processor using the output of each of the four separators on a held out test set (see Section 4.2). Furthermore, to understand whether MSG can also reduce the artifacts produced by an unseen separator, we evaluate our post-processor on the output of a fifth separator that it was never trained on: Hybrid Demucs (v3) [48], which operates in both waveform and spectral domains. For all separation models we used the trained, frozen weights released by the authors, with no alterations. We refer readers to the papers on each separator for architectural and training details.

4.2 Data

All experiments were run with the MUSDB18 dataset [49]. MUSDB18 contains 150 songs: 100 in the training set and 50 in the test set. Each song in MUSDB18 has a full mixture and isolated source audio stems for vocals, bass, drums and a fourth catch-all category called “other”. We omitted this catch-all category because we find that attempting to enhance many instruments at once with the same model greatly increases the difficulty of the task. We performed source separation on every song in MUSDB18 using all five of our source separators, producing source estimates of bass, drums and vocals.

The input audio was peak normalized before passing it through the network. Since Wavenet operates at 16 kHz we use this sample rate. We downsampled all systems to 16 kHz so that there was a uniform sample rate across all separation models. Here, we focus solely on enhancement and leave the task of bandwidth extension for future work. Thus, the output of MSG on all systems was at a sample rate of 16 kHz.

4.3 Training

We trained one MSG model on the MUSDB18 training set for each source class (bass, drums, or vocals). Each model was trained using source estimates from four separators (Demucs v2, Wavenet, Spleeter, and OpenUnmix) as input and the ground-truth sources as training targets. We segmented the audio into 1-second clips and rejected silent clips where the ground-truth source has an RMS below -60dB FS, resulting in over 100,000 training examples per model. On each training iteration, we randomly swapped the input data with ground-truth with a 10% probability. This encourages the model to leave high-quality audio unaltered.

We computed the three resolutions of STFTs passed to our multi-resolution spectrogram discriminators (Section 3) using window sizes of 512, 1024, and 2048 samples and hop sizes of 128, 256, and 512 samples, respectively. We used one Adam optimizer [50] for the generator and another for the discriminator. We used a learning rate of 2e-4 and beta values of $(.5, .9)$. To find suitable loss weights for all 3 types of losses on the generator (LSGAN loss, deep matching loss, and multi-scale spectral loss; see Section 3), we solved a least squares equation to weigh all loss terms equally for the first 1k training iterations. After that, we froze those weights applied to the losses for the remainder of training.

4.4 Subjective evaluation

The goal of our research is to improve the perceptual quality of source separation output. Therefore, we evaluate our MSG post-processor using a crowdsourced subjective evaluation (Section 4.4) rather than reporting an objective metric like Signal-to-Distortion Ratio (SDR) [51,52], since widely-used objective metrics for source separation are imperfect proxies for human perception [34, 52–57].

For evaluation data, we used one seven-second segment from each of the 50 songs from the MUSDB18 test set. We performed source separation on each seven-second segment using each of the five source separation systems (Section 4.1) to create source estimates of bass, drums, and vocals. Each output was then processed with MSG, resulting in 50 matched pairs for each combination of separator and source class: the raw output, and the output processed by MSG.

There are 15 unique combinations of the five separators and three sources (bass, drums, and vocals). For each combination, we performed a two-way forced-choice listening test between the raw output and the output processed by MSG. We initially recruited 20 participants for each test and omitted responses from participants that failed a prescreening listening test. This resulted in a minimum number of 15 participants in any test. Each participant evaluated 25 randomly-selected pairs from the 50 examples for that combination of source and separator.

A two-tailed binomial test was performed where the null hypothesis was that there was no difference between MSG-enhanced and raw separator output. If the results of a particular test showed no difference (i.e., $p < 0.05$) we

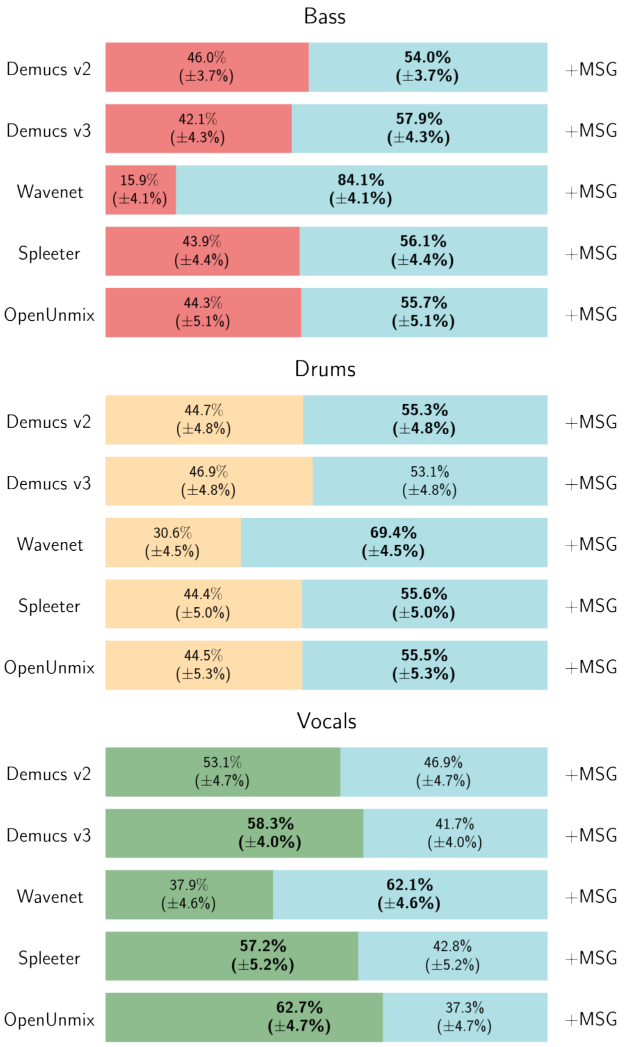

Figure 2: Subjective pairwise test results for bass, drums, and vocals. Each row contains the percent of listeners selecting that option as higher quality in a two-way forced choice listening test. A bold-faced value indicates a statistically significant difference.

recruited an additional 10 participants to see if a difference could be determined.

For each pairwise comparison, participants were given the following instructions, where <*source*> is one of "bass", "drums" or "vocals":

> Listen to both recordings of a <*source*>. After listening to both, select the recording that sounds like a higher-quality <*source*>. The higher-quality recording is the one that is more natural sounding, or has fewer audio artifacts (e.g., noise, clicks, or other instruments).

We used Reproducible Subjective Evaluation (ReSEval) [58] to set up our listener studies. We recruited participants via Amazon Mechanical Turk (MTurk). Our participants were US residents at least 18 years old that completed 20 or more tasks on MTurk with an approval rating of at least 97%. Participants who passed the listening test and completed our evaluation were paid $3.00.

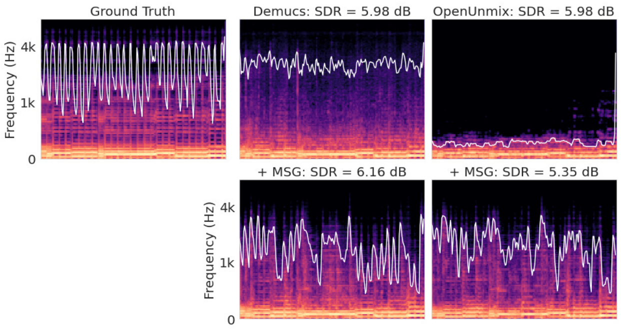

Figure 3: Spectrograms of the ground truth (left) and source estimates from Demucs v2 and OpenUnmix (top) and corresponding MSG output (bottom) for the bass source. The 98% spectral rolloff frequency is overlaid in white.

For each of the 15 tests, we collected between 308 and 696 pairwise evaluations from between 15 and 30 participants who passed the prescreening listening test. The number of evaluations is not a multiple of 25 because a few participants did not finish all 25 examples in their set of pairwise evaluations.

4.4.1 Subjective evaluation results

Each listening test evaluates one combination of source class and source separator. Figure 2 shows the results for each of the 15 tests. Listeners preferred the MSG output to the raw source separator output in 11 out of 15 combinations of separator and source. This difference was statistically significant (using a binomial test) in 10 of the 11 combinations. Listeners preferred the quality of separators with MSG on bass source estimates and had a moderate preference for MSG on drums. For vocals, listeners had a slight preference for the source estimate without MSG.

MSG performed best on the Wavenet separator, where it significantly improved the perceptual audio quality of all sources. MSG was also able to improve on the quality of source estimates of a separator not seen during training, Demucs v3, for bass sources—as well as an improvement on Demucs v3 drum sources that was not statistically significant. Note that Demucs v3 is a hybrid approach that operates in both the waveform and spectrogram domains. Our performance on vocals indicates that MSG is not able to enhance the quality of the source estimate of vocals. We are unaware of prior work that attempts to enhance source separation output, let alone a separated vocal source. Vocal separation has been optimized by many years of existing research, potentially leaving less room for a post-processing system to improve compared to, e.g., drum and bass sources.

5. FURTHER ANALYSIS

Our listener study indicates whether a separation is relatively "good" or "bad", but it does not clarify *why* one sep-

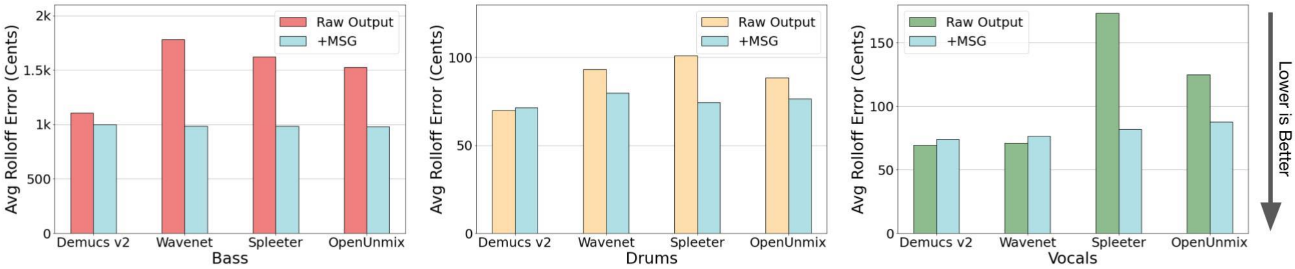

Figure 4: Mean spectral rolloff error for bass, drums, and vocals for separators with and without MSG post-processing.

arated source is better or worse than another. Similarly, the widely-used Signal-to-Distortion Ratio (SDR) [51, 52], as well as the related SIR and SAR, are not designed to capture the specific types of errors we focus on in this work. See, for example, the top row of Figure 3, which shows source estimates produced by Demucs v2 and OpenUnmix for the same bass source from the same mixture. Demucs adds additional high-frequency noise not present in the ground truth, while OpenUnmix removes many of the upper harmonics. Visually, the difference between these two systems is plain, however their SDR values (using `mus_eval` [59]) are equal to two decimal places: *5.98 dB!*

In this section, we examine the output of the four state-of-the-art separation systems used in the training of MSG models: (Demucs v2 [7], Wavenet [8], Spleeter [17], and OpenUnmix [16]), as well as the MSG-processed outputs for those four systems. As before, we use the MUSDB18 [49] test set, and we omit the “other” source.

Anecdotally, we have noted that waveform separators tend to add extra high-frequency noise and spectrogram separators tend to remove high-frequency partials, especially in bass estimates (see Figure 1). Spectrogram separators also tend to smooth out transients. While these are not the only issues that current separation systems exhibit, the rest of this section will be dedicated to analysis of these two issues.

5.1 Added and Missing Frequency Content

Following our anecdotal observation that waveform separators tend to add extra high-frequency noise and spectrogram separators tend to remove high-frequency partials, we seek to formalize these notions.

One statistic that can be a good proxy for whether a source estimate has excess high-frequency content or is missing desirable high-frequency content is spectral rolloff. For a given time frame in a spectrogram, the spectral rolloff at $X\%$ is the frequency below which $X\%$ of the energy of the signal lies. For example, the white line on each spectrogram in Figure 3 shows the spectral rolloff at 98%.

For every song in the MUSDB18 [49] test set, we compute the spectral rolloff at 98% every 32 ms (a hop size of 512 samples at 16 kHz) for every ground truth isolated source, every estimate produced by one of the four training separators and every MSG-enhanced source estimate. To calculate our statistics, we omit any frames that have an RMS less than -40 dBFS, in the ground-truth source, so as not to examine rolloff in relatively silent regions. We report the error between a source estimate's rolloff and a ground-truth source's rolloff in cents, which is $1200 \times (\log_2 x - \log_2 y)$, where x and y are rolloff frequencies in Hz. We chose to use cents over Hz because it better correlates to how humans perceive audio.

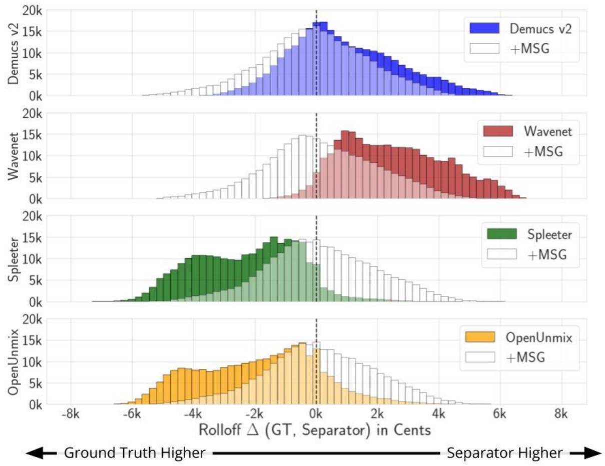

Figure 5: Histogram of the difference in spectral rolloff values between a given separator's bass estimate and ground truth bass source over the MUSDB18 test set. The vertical dotted line shows the desired difference of 0. MSG reduces the difference between the rolloff values of source estimate and ground-truth.

In Figure 4, we show the mean error, in cents, between the ground truth rolloff and the source estimate's rolloff. We see that the source estimates of vocals and drums have spectral rolloff errors on the order 100-200 cents, whereas source estimates of bass have errors of roughly 1000 cents. MSG reduces this error for all separators on bass sources, for three of the four separators on drums, and two of the four separators on vocals.

Because the bass estimates have such large errors, we examine them further in Figure 5, where we show a histogram of the per-frame differences between the spectral rolloff of source estimates and ground truth. The top two rows of the histogram show results for the two waveform separators (i.e., Demucs v2 and Wavenet), which each show an error distribution that is strongly skewed towards positive error. This corroborates our observation of high-frequency noise introduced by waveform separators, as shown in the bass spectrogram in Figure 3. The bottom two rows show the error distribution for two spectrogram separators, Spleeter and OpenUnmix. Both exhibit an error distribution for spectral rolloff that is strongly skewed toward negative values. This quantifies the effect illustrated in Figures 3 and 1, where the spectrogram separators remove the higher partials of the bass source.

We further observe that, when MSG is applied to the output of all four separators, the resulting error distribution is less biased and, as was already shown in Figure 4, reduces the mean error magnitude. Figure 1 illustrates the effect of MSG on a single bass example, showing improved spectral rolloff reconstruction for both waveform and spectrogram models.

5.2 Improving Transient Reconstruction

While listening to the source estimates from spectrogram separators, we noticed that the transients of source estimates for drums and bass did not sound as clear as in the ground truth source estimates. To quantify these observations, we measure the location and strength of onsets in the estimated sources relative to the ground-truth. We use librosa's [60] `onset_strength()` function [61], which computes the spectral flux onset strength envelope at every frame in a spectrogram. We approximate an onset by identifying every frame with a strength above a certain threshold. We select an onset strength threshold via manual tuning. We set the threshold value to a constant value of 0.75 for both bass and drums on the MUSDB18 dataset. We manually tuned this threshold to find a value that best corresponds with our perception of relevant peaks in the signal. We chose to threshold `onset_strength()` instead of using librosa's `onset_detect()` because we found that matching up onsets between two signals using the latter method was hard to correctly tune.

We run this onset strength thresholding on both the ground-truth source and a source estimate and then calculate the F1 between the binary threshold arrays of the raw source estimate and the MSG post-processed estimate as a proxy for how well a separator preserves transients. A true positive (TP) is when a detected onset exists at the same spectrogram frame in the ground-truth and source estimate, a false positive (FP) is when an onset is detected at a frame in the source estimate but not the ground truth, and a false negative is when an onset is detected (FN) at a frame in the ground truth but not in the source estimate. We report the F1 score of onset reconstruction, $TP/(TP + \frac{1}{2}(FP + FN))$.

We report the F1 scores for onset detection on bass, drums, and vocals in Table 1. The results for vocals are not in favor of MSG for 3 of the 4 separators. We include the vocals results for completeness. Evaluating the transients for vocals might be slightly unusual, but the observed results align with the listener studies. In contrast with vocals, bass and drums both see improved F1 scores across multiple separators: MSG improves the F1 score in 7 out of the 8 combinations of source and separator, with the sole exception of drums separated by Demucs v2. This indicates that the ability to represent transients is generally improved by applying MSG-based post processing on bass and drums.

	Model	Type	Onset Strength F1	
			Raw	+ MSG
Bass	Demucs v2	Wave	0.52	**0.54**
	Wavenet	Wave	0.36	**0.44**
	Spleeter	Spec	0.36	**0.52**
	OpenUnmix	Spec	0.39	**0.49**
Drums	Demucs v2	Wave	**0.84**	0.82
	Wavenet	Wave	0.73	**0.74**
	Spleeter	Spec	0.78	**0.82**
	OpenUnmix	Spec	0.78	**0.79**
Vocals	Demucs v2	Wave	**0.58**	0.57
	Wavenet	Wave	**0.51**	0.49
	Spleeter	Spec	**0.71**	0.66
	OpenUnmix	Spec	0.41	**0.57**

Table 1: F1 scores for thresholded onset strength for bass, drums, and vocals for four separators with and without MSG post-processing. "Raw" means that F1 is computed between the separator's raw output and ground truth. "+MSG" means that MSG post-processing is applied to the raw source estimates. According to this measure, MSG is able to better preserve onsets in 7 out of 8 cases between the bass and drums sources, which most clearly demonstrate artifacts with transients.

6. CONCLUSION

State-of-the-art music source separators create audible perceptual degredations, such as missing frequencies and transients. In this work, we propose Make it Sound Good (MSG), a post-processing neural network that leverages generative modeling to enhance the perceptual quality of music source separators. In listening studies, users prefer bass and drum source estimates produced with MSG post-processing—even on a state-of-the-art separator not seen during training. We analyze the errors of waveform-based and spectrogram-based separators with and without MSG. Without MSG, we show that waveform-based separators induce high-frequency noise and spectrogram-based separators fail to reconstruct high-frequencies in the bass source, and have trouble reconstructing transients. We measure these artifacts via spectral rolloff and onset detection and show that, for both bass and drums, MSG generally improves reconstruction of spectral rolloff and onsets of the source estimate relative to the ground-truth sources. Fruitful directions for future work include using more modern techniques for sound enhancement (e.g., diffusion models [43]), making post-processors for the vocals and "other" sources, and deeper analyses of the issues with separation systems.

7. REFERENCES

[1] E. Vincent and X. Rodet, "Instrument identification in solo and ensemble music using independent subspace analysis," in *International Society for Music Information Retrieval (ISMIR)*, 2004.

[2] J. F. Woodruff, B. Pardo, and R. B. Dannenberg, "Remixing stereo music with score-informed source separation." in *International Society for Music Information Retrieval (ISMIR)*, 2006.

[3] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," *Cybernetics & Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[4] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[5] Y. Mitsufuji, G. Fabbro, S. Uhlich, and F.-R. Stöter, "Music demixing challenge 2021," *International Society for Music Information Retrieval (ISMIR)*, 2021.

[6] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[7] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv e-prints*, pp. arXiv–1911, 2019.

[8] F. Lluís, J. Pons, and X. Serra, "End-to-end music source separation: is it possible in the waveform domain?" *Interspeech*, 2019.

[9] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *International Society for Music Information Retrieval (ISMIR)*, 2018.

[10] A. M. Reddy and B. Raj, "Soft mask estimation for single channel speaker separation," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA)*, 2004.

[11] R. J. Weiss and D. P. Ellis, "Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition (SAPA)*, 2006.

[12] Z. Rafii and B. Pardo, "A simple music/voice separation method based on the extraction of the repeating musical structure," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[13] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, "Adaptive filtering for music/voice separation exploiting the repeating musical structure," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[14] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[15] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *International conference on acoustics, speech and signal processing (ICASSP)*, 2017.

[16] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[17] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[18] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *arXiv e-prints*, pp. arXiv–2010, 2020.

[19] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *International Conference on Learning Representations (ICLR)*, 2018.

[20] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "Melgan: generative adversarial networks for conditional waveform synthesis," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[21] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *International Conference on Learning Representation (ICLR)*, 2020.

[22] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *International Conference on Learning Representations (ICLR)*, 2019.

[23] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[24] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Neural Information Processing Systems (NeurIPS)*, 2020.

[25] W. Jang, D. Lim, J. Yoon, B. Kim, and J. Kim, "Univnet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation," *Interspeech*, 2021.

[26] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," *Interspeech*, 2017.

[27] S.-W. Fu, C.-F. Liao, Y. Tsao, and S.-D. Lin, "Metricgan: Generative adversarial networks based blackbox metric scores optimization for speech enhancement," in *International Conference on Machine Learning (ICML)*, 2019.

[28] J. Su, Z. Jin, and A. Finkelstein, "Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks," in *Interspeech*, 2020.

[29] A. Pandey and D. Wang, "Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[30] J. Su, Z. Jin, and A. Finkelstein, "Hifi-gan-2: Studio-quality speech enhancement via generative adversarial networks conditioned on acoustic features," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[31] S.-W. Fu, C. Yu, T.-A. Hsieh, P. Plantinga, M. Ravanelli, X. Lu, and Y. Tsao, "Metricgan+: An improved version of metricgan for speech enhancement," in *Interspeech*, 2021.

[32] P. Andreev, A. Alanov, O. Ivanov, and D. Vetrov, "Hifi++: a unified framework for neural vocoding, bandwidth extension and speech enhancement," *arXiv e-prints*, pp. arXiv–2203, 2022.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Neural information processing systems (NeurIPS)*, 2014.

[34] E. Gusó, J. Pons, S. Pascual, and J. Serrà, "On loss functions and evaluation metrics for music source separation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 306–310.

[35] D. Stoller, S. Ewert, and S. Dixon, "Adversarial semi-supervised audio source separation applied to singing voice extraction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2391–2395.

[36] Y. Li, B. Gfeller, M. Tagliasacchi, and D. Roblek, "Learning to denoise historical music," *International Society for Music Information Retrieval (ISMIR)*, 2020.

[37] E. Moliner and V. Välimäki, "A two-stage u-net for high-fidelity denoising of historical recordings," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[38] J. Imort, G. Fabbro, M. A. M. Ramírez, S. Uhlich, Y. Koyama, and Y. Mitsufuji, "Removing distortion effects in music using deep neural networks," *arXiv preprint arXiv:2202.01664*, 2022.

[39] S. Sulun and M. E. Davies, "On filter generalization for music bandwidth extension using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 1, pp. 132–142, 2020.

[40] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," *arXiv e-prints*, pp. arXiv–1903, 2019.

[41] J. Pons, S. Pascual, G. Cengarle, and J. Serrà, "Upsampling artifacts in neural audio synthesis," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3005–3009.

[42] J. Pons, J. Serrà, S. Pascual, G. Cengarle, D. Arteaga, and D. Scaini, "Upsampling layers for music source separation," *arXiv e-prints*, pp. arXiv–2111, 2021.

[43] N. Kandpal, O. Nieto, and Z. Jin, "Music enhancement via image translation and vocoding," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[44] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *International Conference on Computer Vision (ICCV)*, 2017.

[45] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning (ICML)*, 2016.

[46] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *Interspeech*, 2020.

[47] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.

[48] A. Défossez, "Hybrid spectrogram and waveform source separation," *ISMIR Workshop on Music Demixing (MDX)*, 2021.

[49] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2014.

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

[52] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[53] B. Fox, A. Sabin, B. Pardo, and A. Zopf, "Modeling perceptual similarity of audio signals for blind source separation evaluation," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 454–461.

[54] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman, "Fast and easy crowdsourced perceptual audio evaluation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 619–623.

[55] M. Cartwright, B. Pardo, and G. J. Mysore, "Crowdsourced pairwise-comparison for source separation evaluation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 606–610.

[56] U. Gupta, E. Moore, and A. Lerch, "On the perceptual relevance of objective source separation measures for singing voice separation," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[57] E. Cano, D. FitzGerald, and K. Brandenburg, "Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1758–1762.

[58] M. Morrison, B. Tang, G. Tan, and B. Pardo, "Reproducible subjective evaluation," in *ICLR Workshop on ML Evaluation Standards*, April 2022.

[59] F.-R. Stöter, A. Liutkus, and N. 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.

[60] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Python in Science (SciPy)*, 2015.

[61] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detection," in *Digital Audio Effects (DAFx)*, 2013.

SAMPLEMATCH: DRUM SAMPLE RETRIEVAL BY MUSICAL CONTEXT

Stefan Lattner
Sony Computer Science Laboratories (CSL), Paris, France
stefan.lattner@sony.com

ABSTRACT

Modern digital music production typically involves combining numerous acoustic elements to compile a piece of music. Important types of such elements are *drum samples*, which determine the characteristics of the percussive components of the piece. Artists must use their aesthetic judgement to assess whether a given drum sample fits the current musical context. However, selecting drum samples from a potentially large library is tedious and may interrupt the creative flow. In this work, we explore the automatic drum sample retrieval based on aesthetic principles learned from data. As a result, artists can rank the samples in their library by fit to some musical context at different stages of the production process (i.e., by fit to incomplete song mixtures). To this end, we use contrastive learning to maximize the score of drum samples originating from the same song as the mixture. We conduct a listening test to determine whether the human ratings match the automatic scoring function. We also perform objective quantitative analyses to evaluate the efficacy of our approach.

1. INTRODUCTION

Machine-learning (ML) powered tools are increasingly finding their way into modern music production [1]. Commercial tools already use machine learning for different applications, like mixing/mastering [1] or data visualization. [2] In public perception, ML and artificial intelligence (AI) is often associated with *content generation*. Indeed, also in music production, generative AI models are about to transition from research labs to the studio, in the form of tools that integrate seamlessly into the artist's workflow, like Magenta Studio [3] and others [2–6].

Besides content generation, another frequent task in modern music production is *content retrieval*, where commercial solutions (based on expert systems) also already exist. [4] In manual content retrieval (i.e., content selection), artists select "loops" (short, typically bar-aligned audio files of any content) or "samples" (usually shorter, single notes or drum hits) that fit the current musical context. Content selection is a creative act in its own right, but it is challenging to browse through potentially large loop or sample libraries while – crucially – keeping the current context in mind.

To support the selection process, we envision a retrieval system that learns aesthetic principles of matching drum samples to musical context from data. As a result, it is able to sort a drum sample library according to a *score* it assigns to each sample. This lets users start their search with the most promising options first and may help turn a cumbersome curation process into a more artistic activity.

We use the cosine similarity in a learned latent space as a scoring function. Two encoders are trained with a contrastive loss to maximize the score between mixtures and drum samples originating from the same song. The encoders are trained with an electronic and acoustic data set of songs whose tracks are available separately. We perform a user study to evaluate if human listeners agree with our proposed scoring function. To that end, we test if participants prefer samples that obtain a high rating by the scoring function over randomly selected samples. Furthermore, we perform an ablation study to evaluate the different components of our method. In addition, we want to understand better how the learned space is organized. In particular, we want to gain some insights into the relationship of drum samples that fit well in the same context. For that, we perform a correlation analysis between audio features of samples that are close in the learned latent space.

The paper is organized as follows. Related works are discussed in Section 2, and Section 3 describes the used method. The used data and data preprocessing are described in Section 4. Section 5 explaines how the model training is performed. In Section 6, we introduce the performed experiments, including the objective evaluation, a user study and correlation analysis. We present and discuss the results in Section 7 and conclude in Section 8.

2. RELATED WORK

Self-supervised learning has been a hot topic in recent years, with well-known works in the visual domain, like MoCo [7] and SimCLR [8] that adopt contrastive learning approaches, and BYOL [9], or VICReg [10] that aim to eliminate negative samples. Examples for contrastive approaches in the speech domain are Wav2Vec [11], Wav2Vec 2.0 [12], as well as SpeechSimCLR [13]. Inspired by these works, self-supervised learning in non-speech (i.e., musical and

[1] https://www.izotope.com/
[2] https://algonaut.audio/
[3] https://magenta.tensorflow.org/studio/
[4] https://jamahook.com/

environmental) audio was also introduced with contrastive approaches [14,15], Audio2Vec [16] and Wav2Vec for non-speech audio using a conformer architecture [17]. An example for self-supervised learning for audio based on the BYOL method is [18]. Other audio-related contributions based on contrastive learning are multimodal contrastive learning (for audio and video) [19], as well as multi-format contrastive learning (between raw audio and spectrogram representations) [20]. We adopt ideas from self-supervised learning methods mentioned above, using a dictionary look-up approach like in MoCo [7], we adopt the variance- and co-variance regularization of VICReg [10], and we use decoupled contrastive learning, as proposed in [21].

Other works on drum sample retrieval involve the contrastive learning-based retrieval of single drum samples by their mixed versions [22] and retrieval by vocalization [23, 24]. Other academic works that tackle the task of matching artistic content based on *aesthetic principles* are the Neural Loop Combiner [25], and a stem mashup creation approach [26] (both based on contrastive learning). Other methods for computational mashup creation are based on optimization [27] or expert systems [28].

3. METHOD

3.1 Scoring Function

The goal of our proposed method is to estimate how well a given drum sample fits a specific musical context. This task can be modeled by contrastive learning as a dictionary look-up task, as described in [7]. A musical context $\mathbf{q}_i \in \mathbb{R}^m$ acts as a query, and drum samples $\mathbf{k}_j \in \mathbb{R}^m$ as keys. We define a scoring function $s(\mathbf{q}_i, \mathbf{k}_j)$ that outputs a high score if the sample fits the query. To that end, we use two encoders g_q and g_k to produce a query encoding $\mathbf{u}_i = g_q(\mathbf{q}_i), \mathbf{u}_i \in \mathbb{R}^d$ and a sample encoding $\mathbf{v}_j = g_k(\mathbf{k}_j), \mathbf{v}_j \in \mathbb{R}^d$. Finally, we define the scoring function as the cosine similarity $\mathrm{sim}(\cdot,\cdot)$ between the resulting encodings $s(\mathbf{q}_i, \mathbf{k}_j) = \mathrm{sim}(\mathbf{u}_i, \mathbf{v}_j)$. We train the encoders using a contrastive loss called NT-Xent in [8]:

$$\mathcal{X}(Z) = -\log \frac{\exp\left(\mathrm{sim}(\mathbf{u}_i, \mathbf{v}_j)/\tau\right)}{\sum_{l \neq j} \exp\left(\mathrm{sim}(\mathbf{u}_i, \mathbf{v}_l)/\tau\right)}, \tag{1}$$

where $\{\mathbf{u}_i, \mathbf{v}_j\}$ is a positive pair, $Z \in \mathbb{R}^{n \times d}$ are all representations of a training batch, τ is the temperature parameter, and we adopt the decoupled contrastive learning variant, that has shown to work better for smaller batch sizes, by removing the positive pair from the denominator (i.e., $l \neq j$) [21].

3.2 Regularizations

Furthermore, we combine the contrastive loss with the variance and covariance regularization used in VICReg [10]. The variance regularization term is defined as a hinge function that penalizes variances of latent features along the batch dimension that are smaller than 1 as

$$\mathcal{V}(Z) = \frac{1}{d} \sum_{j=1}^{d} \max\left(0, 1 - S(\mathbf{z}_{:,j}, \epsilon)\right), \tag{2}$$

where Python slicing notation is used, and S is the regularized standard deviation

$$S(x, \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon}. \tag{3}$$

The covariance regularization penalizes non-zero off-diagonal entries in the covariance matrix of each batch, leading to a decorrelation of the latent dimensions:

$$\mathcal{C}(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2, \tag{4}$$

where C is the covariance matrix

$$C(Z) = \frac{1}{d-1} \sum_{i=1}^{d} (\mathbf{z}_{:,i} - \bar{z}_i)(\mathbf{z}_{:,i} - \bar{z}_i)^T, \ \bar{z}_i = \frac{1}{n} \sum_{j=1}^{n} z_{j,i}. \tag{5}$$

Even though the cosine distance renders the norms of the network outputs irrelevant, they can still grow due to optimization dynamics. To keep the norms in a moderate range, we also add a norm regularization in form of a hinge function as

$$\mathcal{N}(Z) = -\frac{1}{n} \sum_{i=1}^{n} \min\left(0, c - ||\mathbf{z}_i||\right), \tag{6}$$

where we fix $c = 4$ in our experiments (because most norms are smaller than 4 at initialization). Putting all the above terms together (and weighting the regularization terms with factors γ, δ and η), yields the final loss

$$\mathcal{L}(Z) = \mathcal{X}(Z) + \gamma\mathcal{V}(Z) + \delta\mathcal{C}(Z) + \eta\mathcal{N}(Z). \tag{7}$$

4. DATA

We used a dataset of electronic music and pop/rock songs of 44.1 kHz sample rate for training and evaluation. The electronic music portion consists of 4830 so-called “remix packs”, with durations ranging from several seconds to several minutes. Each remix pack consists of multiple audio tracks that contain the individual (mutually aligned) instruments (such as synth, bass, guitar, pad, strings, choir, brass, keyboard, vocals, and different percussion instruments).

The Pop/Rock dataset is a proprietary dataset of 885 well-known rock/pop songs of the past decades, including artists such as Lady Gaga, Coldplay, Stevie Wonder, Lenny Kravitz, Metallica, AC/DC, and Red Hot Chili Peppers. The stems (guitar, vocals, bass guitar, keyboard, misc, and different percussion instruments) are available as individual audio tracks for each song.

From every percussion track in the dataset, we extract so-called “one-shots”, single hits with the respective percussion instrument. To that end, we picked those samples where the subsequent onset is of maximal distance (to retain most of the percussion’s potential reverb). From the pop/rock songs, we extract 5 one-shots per percussion track, while from the electronic songs, we extract 1 one-shot per percussion track (as most of the time they are all identical). Altogether, this results in 63042 one-shot drum samples.