Mosaics and collages have been an integral part of art for decades. Particularly important in contemporary media art is the audio mosaic, in which an artist manually combines several audio sources in order to construct one single coherent sound, combining elements from disparate sources. Here we propose an algorithm to automatically create audio mosaics using the simulation-based inference paradigm. Our algorithm takes as input an audio file that one wishes to approximate, and a list of audio files one can use for approximation, finding a posterior distribution from which one can sample reconstructions of the original audio file, using the sources in an interpretable and disentangled manner. We validate our approach by creating an audio mosaic which reconstructs the sound of a traditional Korean funeral using 100 K-pop songs rearranged and overlapped.

Index Terms— machine learning, Bayesian inference, signal processing, probabilistic programming, simulation-based inference, audio mosaic

1. INTRODUCTION

Among post-structuralist texts, the 1980 book “A Thousand Plateaus: Capitalism and Schizophrenia” by Deleuze and Guattari stands out as a seminal experimental work of philosophy, dealing with a wide range of topics originating from the natural world. On the topic of language, the authors state the following:

...relatively few linguists have analyzed the necessarily social character of enunciation...The social character of enunciation is intrinsically founded only if one succeeds in demonstrating how enunciation in itself implies collective assemblages. It then becomes clear that the statement is individuated, and enunciation subjectified, only to the extent that an impersonal collective assemblage requires it and determines it to be so...every statement of a collective assemblage of enunciation belongs to indirect discourse...Direct discourse is a detached fragment of a mass and is born of the dismemberment of the collective assemblage; but the collective assemblage is always like the murmur from which I take my proper name, the constellation of voices, concordant or not, from which I draw my voice [1].

The most natural tool with which a contemporary media artist could express the ideas espoused in this quote would be an audio mosaic, constructing a collective assemblage of voices which materialize into a so-called novel “subjectified enunciation” directly from the aggregate of detached fragments of direct discourse. In more common parlance, this means finding, from a set of source audio files, clips which could be rearranged and overlapped so as to approximate an entirely different target audio file. Such a mosaic would serve as a metaphor for how linguistic meaning is created in a social manner, and that all enunciations of natural language that can be understood must be inherited from other enunciations which others had heard and understood previously, an idea which has resonated throughout Western philosophy for centuries, notably having been used to humorous effect by Humpty Dumpty in Lewis Carroll’s 1871 novel “Through the Looking-Glass” [2] and expanded upon in Wittgenstein’s “Philosophical Investigations” [3] and Davidson’s “A Nice Derangement of Epitaphs” [4].

Previously developed tools for audio mosaicing would be unsatisfactory to artists seeking to realize such concepts. Symbolically, one would wish to create a mosaic out of 1000 simultaneous and overlapping audio sources, as Deleuze had previously claimed that “Being is Voice,” specifically that Being is a “single and same voice for the whole thousand-voiced multiple...a single clamour of Being for all beings” [5] [6]. Prior tools for audio mosaicing required the specification of heavily hand-engineered features and involved intensive human labor in creating the mosaic [7] [8]. Others modified the source signals in unpredictable ways in the process of mosaicing, rendering the results of the mosaic uninterpretable [9]. Lastly, to the best of our knowledge, no prior work has tackled mosaics which overlap several sources of audio at once, as this creates a challenging combinatorial problem out of the scope of previous works.

In order to realize Deleuze’s collective assemblage, we find it natural to look at the audio mosaicing problem from the Bayesian perspective. Recent advances in probabilistic
2. ALGORITHM

Our goal is to create a model specifying how an audio file can be created via combinations of other audio, and then to condition this model on a specific target audio file in order to discover how latent variable draws could have played out in order to realize the target audio. We therefore implemented a simple model which creates an audio mosaic through averaging, with our model’s plate notation shown in Figure 1 and fully specified in Algorithm 1. The algorithm takes as inputs the shape of a target audio file and a list of source audio files preprocessed using the Short-time Fourier transform (STFT) [12], which are converted to amplitudes and dB-scaled relative to a fixed reference. The model starts by initializing an empty audio matrix with $N_{tgt}$ rows and $N_{target}$ columns. The model then draws from a categorical distribution over the STFT segments in the sampled source audio files in order to obtain $N_{clips}$ STFT segments. The model then averages the $N_{clips}$ STFT segments and inserts the average into the audio matrix. The above procedure is repeated for every STFT segment in the audio matrix, filling it out with averaged combinations of STFT segments of the source files iteratively.

Our model, which randomly creates an audio file by selecting and combining STFT segments from a given list of source audio files, can be used in conjunction with approximate inference techniques in order to discover how a specific target audio file could have been created under the model. Specifically, we learn parameters for the categorical distribution segment_samples so that sampling from the model will generate an approximation to an observed target audio file, as opposed to the random combinations of STFT segments generated by the prior. While many choices of approximate inference algorithm are acceptable, we here demonstrate the use of Markov Chain Monte Carlo (MCMC) [13, 10] implemented in PyProb [14, 15]. We chose MCMC specifically due not only to its convergence guarantees, but also because our model of audio was specifically constructed in order to be especially amenable to the single-site versions of MCMC used in PyProb, unlike other models which cannot accept a trace unless multiple latent variable values are changed simultaneously [16].

We can argue that under this model, higher values of $N_{clips}$ specifically make the model more amenable to single-site MCMC. To see this, consider a simplified scenario in which a model draws values from the discrete set $\{1, 4, 6, 8\}$, and averages them to find a posterior over a single observed scalar, 5. If an MCMC chain with $N_{clips} = 2$ is at a state with latent variables 1 and 8 (average 4.5), it could be unable to accept the series of candidates which lead to the optimal latent variables 4 and 6 (average 5) because any single change in latent variables will bring it further from the observation. If one averages many latent variables (large $N_{clips}$), however, this problem disappears, as the model is able to more smoothly...
Algorithm 1: Probabilistic generative model of a target audio file, which is created by selecting and mixing clips from a predefined source collection of audio files.

\[
\begin{align*}
N_{\text{fft}} & \in \mathbb{N}: \text{Length of STFT segments} \\
N_{\text{segments}}^i & \in \mathbb{N}: \text{Number of STFT segments in audio file } i \\
N_{\text{source}} & \in \mathbb{N}: \text{Number of source audio files} \\
N_{\text{clips}} & \in \mathbb{N}: \text{Number of audio clips to mix}
\end{align*}
\]

1: procedure AUDIO_MOSAIC(target_shape, source_audio)
2: \quad \triangleright target_shape is the tuple \((N_{\text{fft}}, N_{\text{target}})\)
3: \quad \triangleright source_audio is a matrix of STFT segments with shape \((N_{\text{fft}}, N_{\text{segments}})\) where \(N_{\text{segments}}^i\) is the concatenation of all STFT segments in all \(N_{\text{source}}\) source files
4: 1
5: 5: Simulate:
6: \quad audio \leftarrow \{\rho_{i,j} = 0\}_{i=1,j=1}^{N_{\text{fft}}, N_{\text{target}}} \quad \triangleright \text{initialize empty reconstruction matrix with } (N_{\text{fft}} \text{ rows}, N_{\text{target}} \text{ cols})
7: \quad \text{for timestep } = 1, \ldots, N_{\text{target}} \text{ do}
8: \quad \quad segment_samples \sim \text{Categorical}(N_{\text{dataset}} \text{ segments}, \text{batch size } = N_{\text{clips}}) \quad \triangleright \text{sample } N_{\text{clips}} \text{ STFT segment indices}
9: \quad \quad audio[\cdot, \text{timestep}] \leftarrow \frac{1}{N_{\text{clips}}} \times \text{sum(audio_clips, axis = -1)} \quad \triangleright \text{put average of audio clips into audio matrix}
10: \quad return audio \quad \triangleright \text{Return reconstructed audio}

coerce the average into becoming closer to the observed 5 despite only being able to change single latent variables at a time. This phenomenon appears empirically as well, with inference problems using low \(N_{\text{clips}}\) having low acceptance rates and failing to converge to a posterior in our more complex but similar model. While increasing the value of \(N_{\text{clips}}\) does not significantly slow down our model due to the fast vectorized sampling of probability distributions available in PyProb, higher values of \(N_{\text{clips}}\) make the inference procedure require more traces to converge to a posterior.

After obtaining a posterior, in order to reconstruct the original signal one must inspect the chosen latent variables, average the corresponding STFT segments, and then perform an inverse STFT operation. This step is necessary as all audio is preprocessed to db-scaled amplitudes before being processed by the model, and thus the original signals cannot be reconstructed from the audio matrix alone. This step relies crucially on the disentangled nature of posteriors obtained through probabilistic programming models, which allows us to tackle a much less computationally intensive inference problem than the alternative, which would involve matching frequency and phase information simultaneously as opposed to amplitudes.

3. EXPERIMENTS

We apply our algorithm in order to do inference on a target audio file, chosen to be a 5 second clip of the sangyeo-sori from a traditional Korean funeral, using source audio chosen to be a collection of 100 full-length K-pop songs from the early 2010s. The traditional Korean funeral typically incorporates traditional Korean instruments such as the haegeum and gayageum which are not found in any of our K-pop dataset, alongside singing at pitches which would be unusual in K-pop, providing a challenging approximation problem.

We note that while MCMC cannot be efficiently parallelized in general, as each STFT segment in the returned audio matrix is independent in our model, one could perform inference using our model with up to \(N_{\text{target}}\) parallel, independent MCMC runs. While in this case one would have to verify the convergence of each of the parallel runs, this strategy would allow for creating mosaics of audio files of indefinite length without necessitating additional wall clock time for inference, given sufficient computational resources.

In order to perform inference, we define a likelihood over the observed (flattened) target audio matrix to be an isotropic multivariate Gaussian with mean equal to a vector containing each target matrix entry and standard deviation which is taken to be a hyperparameter, as is commonly done when using Approximate Bayesian Computation (ABC) techniques [17][18]. While it is possible to do inference with this setup alone using a uniform prior categorical distribution, it is incredibly inefficient, requiring multiple weeks of computation with very large likelihood standard deviations in order for MCMC to converge, even for an audio clip as short as 5 seconds.

In order to overcome the prohibitive time required to perform inference using the uniform prior, in practice we find that using a data-dependent prior, choosing source STFT segments with probability proportional to the cosine similarity to the target STFT segment for that timestep in the categorical draw in Algorithm 1 (line 8), is much more effective. This simple change makes inference in our model more sample efficient by many orders of magnitude, and allows for the use of likelihood distributions with much smaller standard devia-
tion, while still allowing for the inference procedure to model uncertainty around our heuristic-based prior.

Performing random-walk lightweight Metropolis-Hastings MCMC inference \cite{13} with our data-dependent prior gives the convergence and autocorrelation plots shown in Figure 2. The Gelman-Rubin convergence plot \cite{19, 20} shows that MCMC chains with different random seed converge to the same stationary distribution after approximately 100,000 traces (runs of the model) and the autocorrelation plot \cite{21} shows that traces in the MCMC chain approximately 10,000 steps apart are uncorrelated with each other. Re-running MCMC and only taking every 10,000th trace after discarding the first 100,000 will therefore give us samples from the desired posterior.

We show a visualization comparing the spectrogram of the target signal, spectrogram of an average over posterior traces using our data-dependent prior, and the spectrogram of an average over posterior traces using the uniform prior in Figure 3. Doing inference with the uniform prior is computationally inefficient, necessitating the use of likelihood standard deviations so large that nearly all traces are accepted by the Metropolis-Hastings algorithm, and thus posteriors obtained in this fashion cannot closely model the target signal. In contrast, using a data-dependent prior which is biased towards using STFT segments close to the target allows for the use of much smaller likelihood standard deviations, and thus allows for a closer matching to the target signal. We note that the posterior from the data-dependent prior appearing to be a “weaker” or “blurred” version of the target signal is the intended behavior, and reflects the uncertainty of our posterior distribution, as traces from the empirical posterior often conflict with each other. The amount of such uncertainty can be controlled by adjusting the likelihood standard deviation, to the statistician’s preference.

4. CONCLUSION AND FUTURE WORK

In this work, we have presented a simple algorithm for creating audio mosaics using the simulation-based inference paradigm. Our algorithm is nearly fully automated, interpretable, and produces a distribution of audio mosaics for a given target file. We have also explored several tricks for making inference in this model more effective and efficient, allowing for our algorithm to be used in real-world applications with high fidelity. Our model could be easily extended for future work, for instance, by learning how to combine source audio clips rather than simply averaging a fixed number of them. More advanced inference techniques could also be used in order to enable exact inference in our model, which only uses discrete latent variables with repeated structure \cite{22}. Going forward, we plan to use simulation-based inference to create a work of media art, evoking the collective assemblage in Deleuze’s philosophy which served as the motivation for the development of our algorithm.
5. ACKNOWLEDGEMENTS

We would like to thank the Korea Foundation for their generous support for the duration of this project. We would also like to thank Hongseok Yang for his helpful comments on an early draft of this paper.

6. REFERENCES

[1] Gilles Deleuze and Félix Guattari, *Mille plateaux*, Les Éditions de Minuit, 1980.
[2] Lewis Carroll, *Through the Looking-Glass, and What Alice Found There*, Macmillan, 1871.
[3] Ludwig Wittgenstein, *Philosophische Untersuchungen*, Blackwell, 1953.
[4] Donald Davidson, “A Nice Derangement of Epitaphs,” in *Truth and Interpretation*, pp. 433–446. Blackwell, 1986.
[5] Gilles Deleuze, *Logique du sens*, Les Éditions de Minuit, 1969.
[6] Gilles Deleuze, *Différence et répétition*, Presses Universitaires de France, 1968.
[7] Ari Lazier and Perry Cook, “Mosievius: Feature Driven Interactive Audio Mosaicing,” *Audio*, 2003.
[8] Aymeric Zils and François Pachet, “Musical mosaicing,” *Digital Audio Effects (DAFx)*, 2001.
[9] Jonathan Driedger, Thomas Prêtzlitch, and Meinard Müller, “Let it bee – Towards NMF-inspired audio mosaicing,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015*, 2015.
[10] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood, “An Introduction to Probabilistic Programming,” *arXiv preprint*, 2018.
[11] Kyle Cranmer, Johann Brehmer, and Gilles Louppe, “The frontier of simulation-based inference,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30055–30062, 2020.
[12] Ervin Sejdić, Igor Djurović, and Jin Jiang, “Time-frequency feature representation using energy concentration: An overview of recent advances,” *Digital Signal Processing: A Review Journal*, vol. 19, no. 1, 2009.
[13] David Wingate, Andreas Stuhlmüller, and Noah D. Goodman, “Lightweight implementations of probabilistic programming languages via transformational compilation,” in *Journal of Machine Learning Research*, 2011, vol. 15, pp. 770–778.
[14] Atılım Güneş Baydin, Lukas Heinrich, Wahid Bhimji, Bradley Gram-Hansen, Gilles Louppe, Lei Shao, P. Prabhat, Kyle Cranmer, and Frank Wood, “Efficient probabilistic inference in the quest for physics beyond the standard model,” in *Advances in Neural Information Processing Systems*, 2019, vol. 33.
[15] Atılım Güneş Baydin, Lei Shao, Wahid Bhimji, Lukas Heinrich, Lawrence Meadows, Jialin Liu, Andreas Munk, Saeid Naderiparizi, Bradley Gram-Hansen, Gilles Louppe, Mingfei Ma, Xiaohui Zhao, Philip Torr, Victor Lee, Kyle Cranmer, Prabhat, and Frank Wood, “Étalumis: Bringing probabilistic programming to scientific simulators at scale,” in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2019.
[16] Andrew Gambardella, Bogdan State, Naeemullah Khan, Leo Tsourides, Philip H. S. Torr, and Atılım Güneş Baydin, “Detecting and Quantifying Malicious Activity with Simulation-based Inference,” in *ICML Workshop on Socially Responsible Machine Learning*, 2021.
[17] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré, “Markov chain Monte Carlo without likelihoods,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 26, pp. 15324–15328, 2003.
[18] Richard David Wilkinson, “Approximate Bayesian computation (ABC) gives exact results under the assumption of model error,” *Statistical Applications in Genetics and Molecular Biology*, vol. 12, no. 2, pp. 129–141, 2013.
[19] Andrew Gelman and Donald B. Rubin, “Inference from iterative simulation using multiple sequences,” *Statistical Science*, vol. 7, no. 4, pp. 457–472, 1992.
[20] Stephen P. Brooks and Andrew Gelman, “General Methods for Monitoring Convergence of Iterative Simulations,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 434, 1998.
[21] Charles J. Geyer, “Introduction to Markov Chain Monte Carlo,” 2011.
[22] Fritz Obermeyer, Eli Bingham, Martin Jankowiak, Justin Chiu, Neeraj Pradhan, Alexander Rush, and Noah Goodman, “Tensor Variable Elimination for Plated Factor Graphs,” in *International Conference on Machine Learning*, 2019, pp. 4871–4880, PMLR.