EXPLORING LATENT SPACE AUDIO GENERATION FOR DATA SONIFICATION

Ernst STÖTZNER[1] and **Andreas NÜRNBERGER**[1]

[1]*Faculty of Computer Science*, **Otto von Guericke University**, Universitätspl. 2, Magdeburg, 39106 Germany

ABSTRACT

Sonification commonly relies on parameter-mapping techniques that directly translate data values into sound synthesis parameters. In this work, we investigate an alternative approach that utilizes latent space audio generation for data sonification. We propose a method that maps data features to trajectories within the latent space of a pretrained RAVE model, enabling the generation of short audio snippets whose perceptual characteristics are controlled by the data. The approach is evaluated using the Iris dataset as an example for instances and the AirPassenger dataset as a time-series example. To assess the resulting sonifications, distances between Mel-frequency cepstral coefficient (MFCC) derived features are analyzed as a proxy for perceptual similarity. The results indicate that instances from the same class tend to produce more similar audio representations than instances from different classes, suggesting that the data structure is reflected in the generated sound. Furthermore, the time-series sonification demonstrates that temporal changes in the data are translated into perceptible variations in the audio. These findings highlight the potential of latent-space-based audio generation as a flexible approach for data-driven sound design and exploratory sonification.

1. INTRODUCTION

The exploration and interaction with data is a method for gaining insight into patterns and understanding them. To that end, a variety of approaches to information visualization have emerged [1]. In addition to the utilization of visual representations of data, a second method, sonification, has evolved [2]. Sonification involves the representation of data through non-speech sound using a systematic, describable, and reproducible method [3]. While visualization is not accessible to visually impaired individuals, sonification provides an alternative or an augmentation to support their information needs [4].

Lindborg et al. [2] discussed several means and purposes of sonification in their review. The following four points are to be considered: "1) representation and exploration in the classic utilitarian and analytic-oriented way; 2) aesthetics, that is, the role of art, especially electroacoustic music composition and soundscape composition; 3) narration (as a mass medium for communication); and finally, 4) intentionality (What purposes initiate and shape the creative dialogue between sonification-author and sonification-listener?)" [2]. Consequently, it can be utilized for both scientific analysis and the communication of scientific research to a broader audience. A particular sonification design is capable of encouraging users to engage with data and the corresponding topic in greater depth.

In considering the potential of sonification as a data representation tool, we propose a novel approach that utilizes a machine learning model for the generation of sound. Employing the variational autoencoder *RAVE* [5], we are able to regulate the audio output in two distinct manners by latent variables, which are utilized by the decoder for audio generation and by the decoder model itself. This enables our approach to use latent variables for sonification control and allows the decoder model to be replaced for intentionality, aesthetics, or to adapt the sonification to user preferences. Our approach can be classified as a technique of *parameter mapping sonification*, as defined by Hermann et al. [6], but it can also be employed to generate *earcons*.

The objective of this work is to provide an initial foundation for further research and exploration in this evolving field. As a preliminary case study, we applied the proposed sonification to the well-known Iris [7] and Air Passenger [8] datasets. To assess the perceptual distinctiveness of the generated sounds, we compared Mel-Frequency Cepstral Coefficients (MFCCs) [9] of the resulting audio signals. In addition, we examined the acoustic representation of both the time series and the individual data instances, yielding promising results. The sonification scripts, our trained model, and some examples are publicly available:`https://gitlab.com/n-tuc/latent-space-data-sonification`.

In the following, we provide an overview of state-of-the-art sonification tools as well as the use of the *RAVE* architecture. Our specific approach is described in Section 3, including the evaluation methodology. The results are presented in Section 4, followed by a discussion and concluding remarks.

2. RELATED WORK

Sonification has been extensively applied to explore data [2]. To leverage audio in the same way that visualization tools do, several tools have been published in recent years to provide an easy way to sonify scientific data [10–14].

In order to function as a Python visualization library, such as Seaborn [15] or Matplotlib [16], Sonecules [11] was implemented for the purpose of creating sonification

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

pipelines, with the use of commonly used data structures, including NumPy [17] and Pandas [18]. It is claimed that the system fulfills the need to bridge the gap between domain-specific applications and general libraries, serving as a framework for diverse types of data. A similar target followed Strauss [10] to establish a standardized library for sonification. For users with limited programming experience, web frameworks such as Highcharts [13] and Siren [14] offer a more accessible alternative. Rather than proposing a new sonification framework, this work focuses on the parameter-mapping stage and introduces an alternative mapping strategy as a modular component that can be incorporated into existing frameworks. We further assess the potential of the proposed method using the Iris dataset, a dataset that has also been used in earlier sonification examples [19].

Parameter mapping is one of the most widely used sonification techniques [4]. In this work, we propose transferring the mapping of synthesizer parameters, such as pitch, decay, or reverb, to values within a latent space representation. As noted by Hommel and Hermann [20] in their research agenda, latent vectors, among other concepts from the broader field of artificial intelligence, have been identified as a promising direction for future research in sonification. However, their work primarily focused on the use of Large Language Models (LLMs) to support the sonification design process. LLMs have also been explored as a means to support the mapping process itself. Ocampo et al. [21] used an LLM to interpret facility data from a building and translate this interpretation into a predefined set of audio sequences. This approach makes the decision-making process of the LLM observable and analyzable. However, in contrast to traditional parameter-mapping sonification, the resulting audio is not directly driven by the underlying data values but rather by the LLM's interpretation of the data.

The latent space of Variational Auto Encoders (VAEs) has previously been explored for interpolation between audio snippets and for navigating the latent space as a means of musical expression [22–28]. The RAVE architecture [5] has been frequently used in this context due to its streaming capability and its ability to run on standard CPU hardware [24–28]. Since sonification tools are typically expected to operate on conventional machines and provide immediate feedback during user interaction, we also chose to investigate this model. VAEs were also used to explore the latent space of symbolic music [29], but the generated sheet music would be inherently discrete and require a further step for audio synthesis.

Previous work has largely focused on developing interaction methods for navigating the latent space, often requiring the use of example audio material. In contrast, the present contribution focuses on directly controlling the decoder component of the VAE by mapping data into the latent space. In the following section, we introduce our approach for using the RAVE decoder for sonification.

3. METHODS

3.1 Latent Variables Mapping



Figure 1: Overview of the sonification approach with matrix $\hat{Z}$ for controlling the sonification.

The proposed sonification approach uses the decoder component of the RAVE architecture [5] to generate audio directly from latent representations. For the experiments, the publicly available pretrained models *vintage music* and *Musicnet* were used [1]. In addition, we trained a custom model on an artificial music dataset to investigate how musical characteristics such as rhythm, timbre, and melody are represented and organized in the latent space. The controlled structure of this dataset allows us to better analyze how changes in latent variables affect perceptual properties of the generated audio.

The dataset consists of synthesized musical material that includes all minor and major scales, played from low to high pitches and in reverse order, in the form of quarter and eighth notes as well as the semitones of the third intervals of the corresponding chords, including the octave above, arranged over two measures in 4/4 time. For each key, audio snippets were generated at different tempi, with beats per minute ranging from 80 to 160 in five evenly spaced steps. As a synthesizer, the Python package *pyo* [30] was used, combining a sine wave with brown noise to introduce variations in timbre. The influence of the brown noise component was varied from 0 to 40 in five steps, controlling the multiplication factor of the noise density. The combination of these parameters resulted in a dataset of approximately 250 minutes of artificial music. The script used to generate the dataset is also available on GitLab [2].

The RAVE model was trained using the standard configuration parameters for `v1` [3]. However, since the audio material in this dataset is relatively simple, the number of first training phase was reduced to 20,000 and the total number of training iterations to 60,000.

For the sonification framework, the decoder receives a sequence of latent vectors that form the latent matrix

$$\hat{\mathbf{Z}} = \begin{bmatrix} \hat{z}_{1,1} & \hat{z}_{1,2} & \cdots & \hat{z}_{1,T} \\ \hat{z}_{2,1} & \hat{z}_{2,2} & \cdots & \hat{z}_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{z}_{N,1} & \hat{z}_{N,2} & \cdots & \hat{z}_{N,T} \end{bmatrix} \in \mathbb{R}^{N \times T},$$

where N denotes the number of latent dimensions and T

[1] https://acids-ircam.github.io/rave_models_download

[2] https://gitlab.com/n-tuc/latent-space-data-sonification/-/tree/main/artificial_dataset

[3] https://github.com/acids-ircam/rave

the number of temporal steps. Each column therefore represents one latent vector at a given time step, while each row corresponds to the temporal evolution of a latent dimension. The matrix $\hat{Z}$ is provided to the decoder to synthesize the corresponding audio signal, as illustrated in Figure 1.

In the RAVE architecture, each latent vector generates an audio segment of 2048 samples. With a sampling rate of 44,100 Hz, this corresponds to approximately 46 ms of audio. Consequently, generating one second of audio requires approximately 22 latent vectors.

The latent variables learned by the model can be ordered according to their importance for the reconstruction result. To address this objective, following the training of the model, an analysis of the latent representations of the training dataset is conducted using principal component analysis (PCA). This post-hoc analysis produces a ranking of the latent dimensions according to the variance they explain. This ranking is applied to the final models, which are then utilized for the following investigations. As a result, the first latent dimensions capture most of the relevant structure of the audio signal, while later dimensions contribute less information. Consequently, meaningful audio can often be generated using only a subset of the latent dimensions.

To gain further insight into this effect, we evaluated how variations in a single latent dimension influence the output of the decoder. For this purpose, 35 C-major audio snippets were encoded into the latent space Z applying the model trained on the artificial music dataset, and noise was added to a selected latent dimension d: $\hat{z}_{d,t} = z_{d,t} + \mathcal{N}(0, \sigma)$ with $t \in [1, T]$. The noise level was varied across 100 steps with σ ranging from zero to three. For each step, the average MFCC distance between the original audio and the decoded audio was computed to quantify the effect of the added noise. The MFCC-based distance measure used in this evaluation is described in more detail in Section 3.2.1. Each latent dimension was analyzed independently; therefore, the original values of $z_{n,t}$ with $n \neq d$ were kept unchanged during the evaluation. As shown in Figure 2, noise added to the first and second latent dimensions results in the largest changes in the decoded audio. This suggests that these dimensions are the most informative with respect to the audio generation process.

In our approach, the features of the dataset are mapped to the selected latent variables of dimension n. If there are more dimensions in the dataset to be sonified, feature engineering must be performed beforehand. As previously described, it is crucial to map the feature that is hypothesized to have the most information on the first dimensions because it will have the greatest influence on the output. Before mapping the data to the latent variables, each feature is standardized so that the resulting values are centered around zero and have unit variance. This step is important because variational autoencoders are typically trained with a Gaussian prior on the latent space, meaning that latent variables are expected to follow a normal distribution centered around zero. Standardizing the data therefore ensures that the mapped values lie within a range that is consistent

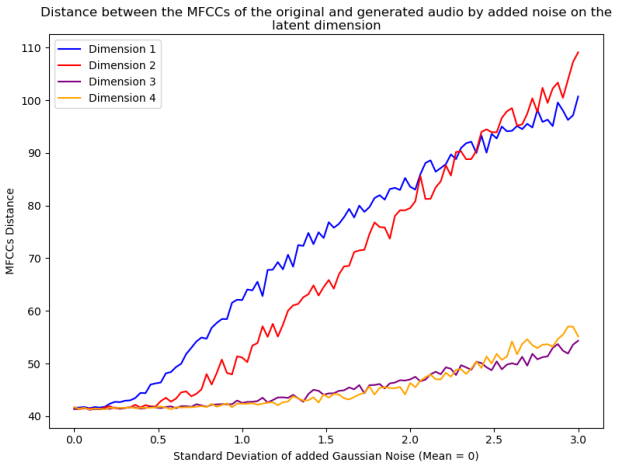


Figure 2: The influence of noise on the latent values of a given dimension.

with the latent distribution learned during training. The resulting latent matrix $\hat{Z}$ is then passed to the RAVE decoder to generate the corresponding audio waveform, as illustrated in Figure 1.

3.2 Case Study

3.2.1 Instances of Iris Dataset

Motivated by the observation of encoded latent vectors from audio snippets of the artificial music dataset, we identified that temporal variation of $\hat{z}_{1,t}$ is particularly important for generating melody-like structures. However, when generating audio from a single instance, such as those in the Iris dataset [7], which describes three classes of flowers based on sepal and petal measurements (width and height), no inherent temporal progression is present in the data. Therefore, a temporal evolution in the latent representation had to be introduced.

To address this, we introduced controlled sign changes in the first two latent dimensions, since these were observed to have the strongest influence on the decoder output. We propose that, the interval between sign changes depends on the min–max normalized value of *petal_length* p_l. This feature is particularly suitable, as it already provides a strong separation between one class and the other two classes in the dataset. The maximum number of potential steps is defined as max_s. The step size that determines how long a latent value remains unchanged in the trajectory is computed as $\max(1, p_l \cdot max_s)$, where $max_s = 10$ was used in the experiments. Consequently, instances with longer petals result in larger step sizes, meaning that the latent values remain constant for longer intervals and therefore fewer sign changes occur over time. In contrast, shorter petals lead to smaller step sizes and more frequent changes in the latent trajectory.

The four dataset features were mapped to the latent dimensions as follows. Since *petal_width* and *petal_length* are correlated, a sepal feature was mapped to the second latent dimension in order to encode complementary information in the generated sound:

1. $\hat{z}_{1,t}$:= standardized *petal_length*, $t \in [1, T]$
2. $\hat{z}_{2,t}$:= standardized *sepal_width*, $t \in [1, T]$
3. $\hat{z}_{3,t}$:= standardized *sepal_length*, $t \in [1, T]$
4. $\hat{z}_{4,t}$:= standardized *petal_width*, $t \in [1, T]$

For the experiments, the latent representation matrix $\hat{Z}$ was constructed with $T = 80$ time steps and $N = 4$ latent dimensions. Using a sampling rate of 44,100 Hz and generating 2,048 samples per time step results in audio snippets with a duration of approximately four seconds. The resulting latent trajectories were then decoded by the RAVE model to generate the final audio representations. This mapping strategy is not limited to the Iris dataset. Similar strategies could be applied to other datasets by assigning their features to latent dimensions and defining appropriate temporal trajectories in the latent space.

To evaluate the generated audio, we measure the distance between Mel-frequency cepstral coefficient (MFCC) representations of the audio snippets. MFCCs are widely used in audio analysis because they approximate aspects of human auditory perception by representing the spectral envelope on a perceptually motivated Mel scale [9, 31]. While originally developed for speech processing, MFCC-based representations have also been successfully applied to music similarity [9, 32]. In this study, MFCCs are used as an objective proxy for perceptual similarity between sonification outputs. The underlying assumption is that changes in the mapped data features should result in measurable differences in perceptually relevant spectral characteristics of the generated audio. Consequently, distances in the MFCC feature space serve as an indicator of how well the parameter mapping reflects variations in the data and how distinguishable instances from different classes may be for listeners.

MFCCs were computed using the Python library *librosa* [33] with a sampling rate of 44.1 kHz, matching the generated audio. Following the approach used for music similarity analysis in [32], we extract 12 MFCC coefficients. For each coefficient, the mean and standard deviation are calculated across all analysis frames, resulting in a descriptive feature vector of length 24 for each audio snippet. This representation summarizes the sound's overall spectral characteristics. Calculating the Euclidean distance between their corresponding feature vectors is called the MFCC distance in the following.

3.2.2 Air Passenger Time Series

As an application of the proposed sonification approach to time-series data, we used the well-known Air Passenger dataset [8]. This dataset contains the monthly number of passengers of a US airline between 1949 and 1960. Since time-series data inherently represents a sequence of values over time, it was not necessary to simulate temporal variation through oscillating control signals. Instead, the temporal structure of the dataset was directly mapped to the latent representation.

The length T of the latent representation was therefore determined by the number of months in the dataset. The standardized number of passengers at month t was mapped to the corresponding latent variable $\hat{z}_{1,t}$, which controls the audio generation over time. All other latent variables were set to zero. In theory, these remaining latent variables could be used to represent additional features when sonifying datasets with higher dimensionality using the latent space of the RAVE model.

4. RESULTS

Table 1: MFCC-Based Distance Analysis of Sonified Audio Instances

Model	**Average Distance**	**Intra-Class Distance**	**Inter-Class Distance**
Artificial Music	76.6±38.6	41.5±19.9	94.5±22.5
Vintage Music	79.5±37.6	48.4±21.8	95.5±25.8
Musicnet	120.7±68.7	64.9±30.1	149.2±43.9

Table 1 presents an analysis of the average distances between the generated audio snippets produced by the proposed sonification approach. The distances are computed between MFCC feature representations of the generated audio signals, as described in Section 3.2.1. We compare intra-class distances, i.e., distances between audio snippets generated from instances belonging to the same class, and inter-class distances, i.e., distances between instances from different classes. The results indicate that the generated audio tends to be more similar within classes and more distinct between classes.

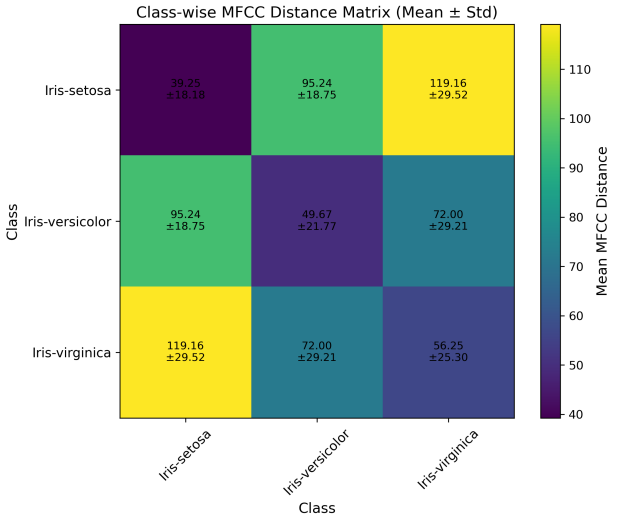


Figure 3: The average distances between the MFCC features of the sonified instances by class. Sonification was done by the model that was trained on vintage music.

Figure 3 shows the intra-class and inter-class distances for each class for the model trained on the vintage music dataset. For all classes, the average intra-class distance is lower than the average distance to instances from other

classes. When comparing the classes, it can be observed that the class *iris-setosa* has the largest average distance to the other classes, indicating that the generated audio for this class is more distinct than for *iris-versicolor* and *iris-virginica*. In contrast, the relatively small average distance between *iris-versicolor* and *iris-virginica* suggests that the corresponding sonifications are more similar to each other. Similar patterns can be observed for the models trained on the other datasets and generally reflect the distribution of the instances according to their classes in the original feature space.

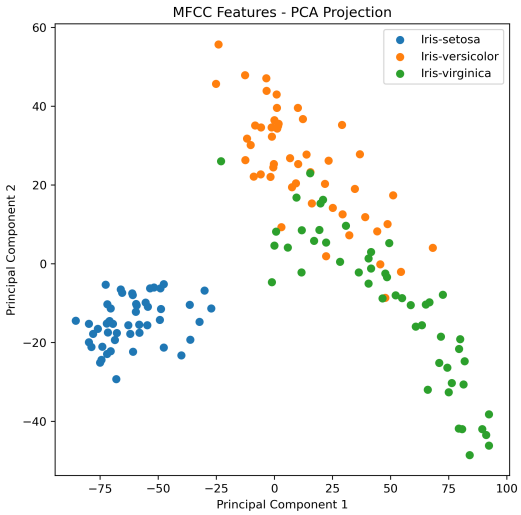


Figure 4: The MFCC features of the audio generated by the model trained on *vintage music* projected onto two dimensions by PCA.

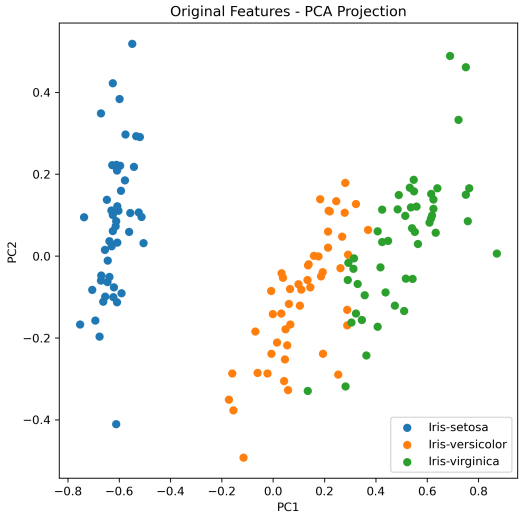


Figure 5: The original iris features projected onto two dimensions by PCA.

To further investigate these results, we applied principal component analysis (PCA) [34] to the 24-dimensional MFCC feature vectors and reduced them to two principal components. The projected audio instances are visualized as a scatter plot with their corresponding classes in Figure 4. For comparison, Figure 5 shows the projection of the original dataset features into two principal components using PCA. Although PCA is applied to different feature representations in each case, the results reveal a similar overlap between the feature distributions of *iris-versicolor* and *iris-virginica*.

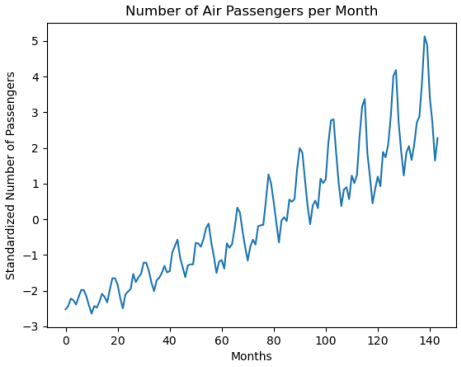


Figure 6: Air passenger data line chart.

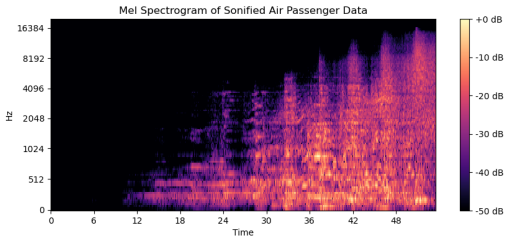


Figure 7: Mel spectrogram of the corresponding sonification created by the *vintage music* decoder.

Figure 6 shows the time-series data that was sonified using the proposed approach described in Section 3.2.2. As illustrated in Figure 7, which presents the Mel spectrogram of the audio generated by the *vintage music* decoder, the increasing trend of the time series is reflected in the spectral evolution of the sound. This indicates that changes in the underlying data are translated into corresponding changes in the audio signal, making the temporal structure of the data perceptually accessible.

Listening to the generated audio reveals noticeable variations in perceived loudness and timbre over time, forming wave-like transformations that evolve from quieter, string-like textures to more vocal-like tones. These perceptual changes correspond to the wave-like progression of the data values and demonstrate how trajectories in the latent space influence the resulting sound. Similar behavior was observed for the other decoder models, which also generated audio that gradually evolves from lower values at the beginning of the time series.

5. DISCUSSION & OUTLOOK

The sonification results presented in Section 4 for the AirPassenger time series show promising outcomes. However, the current approach is limited to a single dimension and should be further evaluated on multidimensional datasets

such as the Iris dataset. Based on the results obtained from the Iris dataset sonification, it can be assumed that variations in features within a time series would likewise lead to distinguishable audio characteristics. The time series sonification approach introduced in this work could, for example, be applied to engage users with climate data, similar to the parameter-mapping approach presented by Van Nispen and Rebel [35]. Our method offers the possibility to control the audio generation directly through the data, while also allowing the overall sound design to be adapted by training the RAVE model on topic-related audio material, such as recordings of rain, wind, or fire.

In prior work, the sonification of the Iris dataset was presented as a single audio snippet representing the entire dataset[4] [19]. While the method described in Section 3.2.1 generates audio snippets on a per-sample basis, feature variations could instead be mapped onto latent variables, as proposed for time series in Section 3.2.2. This would enable the generation of a single audio snippet representing the complete dataset.

In both case studies, only one or four latent dimensions were used for the audio decoder. If higher-dimensional data need to be sonified, either a mapping strategy combining features needs to be engineered, or dimension reduction must be applied beforehand.

The results in Section 4 suggest that the sonifications generated from the Iris instances form class-specific structures in the MFCC feature space assessed through distance calculation, indicating that the generated audio corresponding to different classes may be distinguishable. This observation is further supported by the distribution of instances in the MFCC feature space, which shows a similar structure to the distribution in the original feature space. This suggests that the generated audio preserves relevant characteristics of the original data features.

Consequently, these results indicate that the data features effectively control the audio generation process, and that variations in feature values and their combinations are reflected in the resulting sounds. In other words, changes in the underlying data appear to translate into perceptible differences in the generated audio.

Although the MFCC-based evaluation suggests that the audio samples may be distinguishable to human listeners, this assumption has not yet been validated through a perceptual user study. Future work should therefore investigate how well these results align with human perception by conducting listening experiments.

Due to the independence of the proposed sonification approach from the training dataset used for the audio generator, it offers opportunities for more user-centric sonification design in future work. For example, users could switch between different decoder models to explore alternative auditory representations of the same data. In this context, it would be interesting to investigate how the approach could be used to represent instances of a dataset through topic-related auditory cues.

Short audio cues, such as earcons [36] or auditory landmarks [37,38], have been widely used in user interface design to support orientation and navigation. Since the proposed sonification approach generates short audio snippets, it has the potential to contribute to the design of such cues and to enable more flexible and user-centered auditory interfaces.

Furthermore, the proposed method could be integrated into multimodal data exploration frameworks that combine visual and auditory representations, such as SoniScope [12]. In such settings, latent-space-based audio generation could complement visual analysis by providing an additional perceptual channel for identifying patterns, relationships, or changes in the data.

[4] `https://sonification.de/handbook/chapters/chapter8/`, accessed on March 19, 2026.

6. CONCLUSION

This study examined the application of latent-space audio generation in data sonification, mapping data features to the latent space of a pretrained RAVE model. The results show that the proposed approach can produce audio representations that reflect structural properties of the underlying data. In the experiments with the Iris dataset, instances belonging to the same class resulted in similar MFCC feature patterns, while instances from different classes produced more distinct audio representations. Additionally, the time-series example demonstrates that variations in the data can be translated into perceptible temporal changes in the generated sound.

These findings indicate that latent-space-based audio generation provides a flexible framework for data-driven sound design and sonification. However, the current evaluation relies on MFCC-based similarity measures as a proxy for perceptual differences. While this provides an objective indication that the generated audio reflects the structure of the data, future work should include perceptual studies to assess how well these differences are perceived by human listeners.

A key aspect of the proposed approach is its independence from the dataset used to train the audio generator. This allows different decoder models to be used to create alternative auditory representations of the same data, enabling flexible and potentially user-centered sonification design. Such properties may be particularly relevant for exploratory data analysis and multimodal data exploration systems.

7. REFERENCES

[1] D. Ceneda, T. Gschwandtner, and S. Miksch, "A review of guidance approaches in visual data analysis: A multifocal perspective," *Computer Graphics Forum*, vol. 38, no. 3, pp. 861–879, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13730

[2] P. Lindborg, V. Caiola, P. Ciuccarelli, M. Chen, and S. Lenzi, "Re(de)fining sonification: Project classification strategies in the data sonification archive," *Journal of the Audio Engineering Society*, vol. 72, no. 9, pp. 585–602, Apr. 2024.

[3] K. Liew and P. Lindborg, "A sonification of cross-cultural differences in happiness-related tweets," *AES: Journal of the Audio Engineering Society*, vol. 68, no. 1/2, pp. 25–33, Jan. 2020.

[4] K. Enge, E. Elmquist, V. Caiola, N. Rönnberg, A. Rind, M. Iber, S. Lenzi, F. Lan, R. Höldrich, and W. Aigner, "Open your ears and take a look: A state-of-the-art report on the integration of sonification and visualization," *Computer Graphics Forum*, vol. 43, no. 3, p. e15114, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.15114

[5] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *CoRR*, vol. abs/2111.05011, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[6] T. Hermann, A. Hunt, and J. G. Neuhoff, *The Sonification Handbook*. Berlin: Logos Verlag, 2011.

[7] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x

[8] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day, 1970, first Edition.

[9] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *International Society for Music Information Retrieval Conference*, 2000. [Online]. Available: https://ismir2000.ismir.net/papers/logan_paper.pdf

[10] J. W. Trayford and C. M. Harrison, "Introducing strauss: A flexible sonification python package," in *Proceedings of the International Conference on Auditory Display (ICAD)*. Georgia Institute of Technology, 2024.

[11] D. Reinsch and T. Hermann, "Sonecules: A python sonification architecture," in *Proceedings of the 28th International Conference on Auditory Display (ICAD)*. Norrköping: ICAD, 2023.

[12] K. Enge, A. Rind, M. Iber, R. Höldrich, and W. Aigner, "Towards multimodal exploratory data analysis: Soniscope as a prototypical implementation," in *24th Eurographics Conference on Visualization, EuroVis 2022 - Short Papers, Rome, Italy, June 13-17, 2022*, M. Agus, W. Aigner, and T. Höllt, Eds. Eurographics Association, 2022, pp. 67–71. [Online]. Available: https://doi.org/10.2312/evs.20221095

[13] S. Cantrell, B. Walker, and O. Moseng, "Highcharts sonification studio: An online, open-source, extensible, and accessible data sonification tool," in *Proceedings of the International Conference on Auditory Display (ICAD)*. Georgia Institute of Technology, 06 2021, pp. 210–216.

[14] T. Peng, H. Choi, and J. Berger, "Siren: Creative and extensible sonification on the web," *Proceedings of the 28th International Conference on Auditory Display (ICAD2023)*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264138064

[15] M. L. Waskom, "Seaborn: Statistical data visualization," 2021. [Online]. Available: https://seaborn.pydata.org

[16] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[17] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[18] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference (SciPy)*, 2010, pp. 56–61.

[19] S. Ferguson, W. L. Martens, and D. Cabrera, "Statistical sonification for exploratory data analysis," in *The Sonification Handbook*, T. Hermann, A. Hunt, and J. G. Neuhoff, Eds. Berlin, Germany: Logos Publishing House, 2011, ch. 8, pp. 175–196.

[20] F. Hommel and T. Hermann, "Ai meets sonification: Research agenda and technology demonstration," in *Proceedings of the 30th International Conference on Auditory Display (ICAD)*, 07 2025, pp. 56–63.

[21] R. Ocampo, O. R. Bown, B. Wright, J. Shave, and C. Pegram, "Interpretative data sonification: Using llms to interpret data and generate continuous soundscapes at the sydney opera house," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Porto, Portugal, 2024, pp. 102–109. [Online]. Available: https://smcnetwork.org/smc2024/papers/SMC2024_paper_id243.pdf

[22] G. Le Vaillant and T. Dutoit, "Latent space interpolation of synthesizer parameters using timbre-regularized auto-encoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3379–3392, 2024.

[23] K. Tatar, K. Cotton, and D. Bisig, "Sound design strategies for latent audio space explorations using deep learning architectures," in *Proceedings of Sound and Music Computing*, 2023.

[24] T. Khallaghi, P. Bennett, and A. Tanaka, "Squishysonics: A deformable interface for the physical control of real-time ai sound generation tools," in *Proceedings of the Extended Abstracts of the CHI Conference*

on Human Factors in Computing Systems, ser. CHI EA '25. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3706599.3721175

[25] B. K. Bacon, L. Auri, and H. A. McKee, "Approaches to notation for embodied engagement with a novel neural network-based musical instrument," *Frontiers in Computer Science*, vol. Volume 7 - 2025, 2025. [Online]. Available: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1597806

[26] K. Tahiroglu and L. Wyse, "Latent spaces as platforms for sonic creativity," in *Proceedings of the 16th International Conference on Computational Creativity*, 2024.

[27] F. Liu, L. Blanchard, D. D. Haddad, and J. Paradiso, "Two sonification methods for the mindcube," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Canberra, Australia, June 2025. [Online]. Available: http://nime.org/proceedings/2025/nime2025_74.pdf

[28] M. H. Valenzuela and E. Tomás, "Nebula: A pca-based method to explore rave-encoded audio representations," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Graz, Austria, 2025, pp. 49–56. [Online]. Available: https://doi.org/10.5281/zenodo.15839719

[29] N. Carvalho and G. Bernardes, "Exploring sampling strategies in latent spaces for music generation," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Porto, Portugal, 2024. [Online]. Available: https://hdl.handle.net/10216/165168

[30] O. Belanger, "Pyo, the python dsp toolbox," in *Proceedings of the 24th ACM International Conference on Multimedia*, ser. MM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1214–1217. [Online]. Available: https://doi.org/10.1145/2964284.2973804

[31] I. McLoughlin, L. Pham, Y. Song, X. Miao, H. Phan, P. Cai, Q. Gu, J. Nan, H. Song, and D. Soh, "Spectrogram features for audio and speech analysis," *Applied Sciences*, vol. 16, no. 2, p. 572, 2026. [Online]. Available: https://doi.org/10.3390/app16020572

[32] J. Foote, "A music similarity function based on signal analysis," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 1997, pp. 464–467.

[33] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference (SciPy)*, 2015, pp. 18–25.

[34] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, 2002.

[35] T. van Nispen and K. T. Rebel, "Harmony in complexity: A musical data sonification of one century of climate data," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Graz, Austria, 2025, pp. 86–91. [Online]. Available: https://doi.org/10.5281/zenodo.15839861

[36] T. Dingler, J. Lindsay, and B. Walker, "Learnability of sound cues for environmental features: Auditory icons, earcons, spearcons, and speech," *Proceedings of the 14th International Conference on Auditory Display, Paris, France*, vol. 2008, 2008.

[37] B. Gao, Z. Chen, X. Chen, H. Tu, and F. Huang, "The effects of audiovisual landmarks on spatial learning and recalling for image browsing interface in virtual environments," *Journal of Systems Architecture*, vol. 117, p. 102096, 2021.

[38] E. Stötzner, J. Höbel-Müller, and A. Nürnberger, "Listen! audible landmarks for exploratory information retrieval in virtual reality: Poster track," in *2025 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2025, pp. 187–191.