Brave: Designing an Embedded Network-Bending Instrument, Manifesting Output Diversity in Neural Audio Systems

Daniel Manz and **Mick Grierson**
Creative Computing Institute
University of Arts London
United Kingdom
d.manz@arts.ac.uk
m.grierson@arts.ac.uk

Abstract

As neural audio synthesis becomes more widely adopted there is a growing risk that its limitations could impact the content, quality and diversity of music. Some musicians, artists, and researchers perceive an increased risk of cultural homogenisation and qualitative degeneration due to poor-quality training data and parameterisation. This work seeks to explore new methods for addressing these challenges by contributing to the developing field of "network-bending". Network-bending employs direct manipulation of internal ML architectures to enable active divergence from the training corpus, increasing the statistical variability and capability of model outputs. We present "Brave": an embedded, network-bending hardware instrument, which can provide a novel blueprint for embedding a network-bending system on a stand-alone system. Through a process of iterative musician-led feedback, drawing on Proof-of-Concept Media and Arts Technology approaches, this work seeks to stimulate futher interest in network-bending frameworks applied in the field of AI-driven sound synthesis.

Introduction

Generative AI has pushed the boundaries of artistic creation - It can automate laborious aspects of the artistic process, act as an initial idea generator, and democratise expression. However, this technology bears a critical potential downside, in that it could homogenise the cultural landscape, simplifying the process of generating statistically average output. Therefore, as these tools are further embedded into society and artistic practice, active divergence from their monolithic output may become increasingly important.

This work contributes to the "network-bending" framework - the direct manipulation of neural network parameters in deep generative modelling. Current network-bending technologies require specialist knowledge to operate, demanding a steep learning curve for novel users. In addition, they often require access to high-performance GPUs for real-time performance. This project aims to address these research gaps by leveraging a user-centred design approach and providing a novel blueprint for embedding a network-bending system on a stand-alone system.

Through the fabrication of a physical, network-bending instrument, this work aims to benefit musicians wishing to integrate alternative ML methods into their practice. The instrument, if successful, could aid in enabling greater participation in network-bending technologies and contribute to inspiring further investigation within music technologist and sound art communities.

Related Works

This work builds upon prior research in two key areas. Firstly, we review the technical basis of RAVE-based neural audio synthesis. Secondly, we situate Brave within the broader context of the network-bending framework being used in the creative space, highlighting key influences for the development of the instrument.

Neural Audio Synthesis

Whilst discriminative modelling (Ng and Jordan 2001) provided the bulk of early ML research, in the last decade there have been leaps forward through novel applications of deep learning to generative modelling tasks. Generative models are now capable of synthesising realistic sound both in the frequency domain (Vasquez and Lewis 2019), audio domain (van den Oord et al. 2016) and images (Karras et al. 2021).

Auto-Encoder (AE) (Hinton and Salakhutdinov 2006) architecture is composed of an encoder and decoder. The encoder compresses high-dimensional input data into a lower-dimensional representation vector - known as the latent vector z. The decoder decompresses this latent representation back to the original domain. This architecture exploits the manifold hypothesis (Fefferman, Mitter, and Narayanan 2013): high-dimensional data exhibits an intrinsic low-dimensional structure, despite being embedded in high-dimensional space. The low-dimensional embedding is defined as the latent space. Variational Auto-Encoders (VAEs) (Kingma and Welling 2014) regularise the latent space with parameters mean (μ) and standard deviation (δ). Becoming one of the most well-known deep learning generative architectures, they have become widely used in text (Wang et al. 2019; Semeniuta, Severyn, and Barth 2017), image (Vahdat and Kautz 2020; Huang et al. 2018) and audio synthesis (Haque, Rana, and Schuller 2020).

Realtime Audio Variational autoencoder (RAVE) (Caillon and Esling 2021) leverages VAE architecture for fast and high-quality audio waveform synthesis. The model is able to generate 48 kHz audio in real-time on a standard laptop

CPU. The RAVE model employs a two-stage training procedure. A perceptually motivated reconstruction objective is introduced as "multiband spectral distance". Once this metric converges, adversarial fine-tuning is leveraged with the aim of improving audio quality/naturalness. When the learned representation is considered sufficient for use, the encoder is frozen. Audio signals can then be synthesised and fed to the discriminator, optimising the adversarial objective.

Network-bending Framework

Terence Broad et al. were the first to introduce the notion of "network-bending" (Broad, Leymarie, and Grierson 2021). The authors introduce a new framework for the direct manipulation of deep generative models, facilitating active divergence from the original training dataset. Deterministic transformations are formulated, which can be inserted into the computational architecture of a trained generative neural network and applied during inference. This methodology was applied to both StyleGAN2, for image generation, and a VAE trained on spectrograms of music samples.

The network-bending framework can be placed upon a broader cultural timeline, extending a practice of technological augmentation, subversion and ownership. The name inherits from "circuit bending", coined by Qubais Reed Ghazala (2004), denoting the practice of modifying low voltage electronic devices through the removal/addition of electronic components. Ghazala compares the practice to the creation of an instrument from a "coconut washed up on the shore" - augmenting society's electronic refuse to fabricate a unique sound generator. In its relation to DIY culture and finding chaos within the mass-produced norm, Brave seeks to draw connections between "network-bending" and Ghazala's practice.

Methodology

Framework

The study was guided by the Proof-of-Concept (PoC) implementation of Media and Arts Technology (MAT) studies (Bryan-Kinns and Reed 2023). This implementation provides a blueprint for the design/evaluation of novel interactive systems. A user-study was conducted - adopting an exploratory approach, prioritising open-ended, playful interaction and qualitative feedback over quantitative hypotheses. The PoC research question, which was explored in the user-feedback cycles, was formulated as follows.

What if I surface internal neural network parameters with a physical interface in the context of audio synthesis? How do artists/technologists respond to this, and what playful interactions/technical insight can be prompted?

Participants

6 participants participated in the user-study, sampling a demographic of diverse musical/technical backgrounds aligning with the target demographic. Demographic data is shown in Table 1.

Study Procedure

The study was conducted in a home studio, equipped with Audio-Technica ATH-M50x headphones that provided audio playback during the study. This took place over two weeks. At the beginning of each session, participants signed a consent form and completed a demographic questionnaire. A technical and conceptual overview of the project was provided, assuming no specialist knowledge - this included instructions on how to operate the instrument. They were then prompted to begin their interaction, which was video recorded for retrospective analysis. Then, a series of open-ended questions, gauging both technical/conceptual experience of the user, were asked in interview form and later transcribed for analysis. The questionnaire prioritised open-ended, playful interaction and qualitative feedback over quantitative hypotheses. The questionnaire participants were given can be found at the following link: `https://danielmanz17.github.io/Brave-Questionnaire`.

Embedded Network-Bending

We leverage Kotowski's network-bending fork [1] of the IRCAM $\tilde{nn}$ repository. The decision was made to compile this work and embed it within a stand-alone instrument. The work explores how deep learning architectures can be embedded on a stand-alone device, integrating network-bending, and to further understand how this framework can be used as an "instrument". It is hoped that this can be scaled, motivated by greater efficiency and reduced form factor as parallel processor architecture continues to advance. As this is a PoC, a Raspberry Pi 5 was chosen as

[1] `https://github.com/blazejkotowski/nn_tilde_bending`

Table 1: Demographic Data of Participants in Brave User Study

Participant	Age	Traditional music Experience	Electronic music Experience	Technical Proficiency	ML familiarity
P1	36	Advanced	Advanced	Expert	Advanced
P2	24	Beginner	Beginner	Advanced	Advanced
P3	26	Beginner	Advanced	Advanced	Intermediate
P4	32	Beginner	Beginner	Advanced	Intermediate
P5	24	No experience	Beginner	Advanced	Intermediate
P6	27	Advanced	Advanced	Advanced	Advanced

the platform for this work. A full script, which can be used on other devices, can be found in the Brave repository [2].

Technical Design and Implementation

The Raspberry Pi 5 has a 2.4 GHz quad-core 64-bit Arm Cortex-A76 CPU, a significant upgrade from the Raspberry Pi 4, and a potential 8 GB of RAM. A fan was installed as an active cooling mechanism to prevent thermal throttling - the patch is highly intensive on the CPU/RAM. The Raspberry Pi 5 DAC+3 [3] was used. This has the Texas Instruments PCM5122 DAC delivering stereo analogue audio to a dedicated headphone amplifier, supporting 24-bit 192 kHz high-resolution digital audio. The DAC uses pass-through pins, meaning the peripheral interface elements were still able to attach to the GPIO pins.

System Architecture

A networking solution was used to communicate between peripheral interface elements and Pure Data. OSC receives signals from five digital encoders which are mapped to network-bending parameters, and a switch which can reset the entire model or individual layer weights. The Python script can be found in the Brave repository.

The Adafruit I2C QT Rotary Encoder [4] breakout was used with a standard 24-pulse encoder. This uses the I2C Stemma QT communication protocol, a variant of the I2C protocol developed by Adafruit. The protocol uses JST SH 6- pin connectors and cables, allowing easy daisy-chaining of multiple devices without soldering. This reduces wiring complexity and minimises the use of GPIO pins. A momentary button switch was connected to pins 13 and 14 (GND/5V). Finally, a Waveshare 4.3-inch DSI capacitive touchscreen LCD [5] was attached via the 4-lane MIPI DSI/CSI connector - capacitive was chosen over resistive due to sensitivity/UX considerations.

Design Elements

Physical housing was designed using the Computer-Aided Design (CAD) software Autodesk Fusion 360 and 3D printed. The original 3D print used white filament, which was later painted. A small rectangle was debossed to place the Brave "logo". This used an old Germanic "Rundgotisch" font.

A simple ensemble of visual labels for the four primary encoders was introduced to indicate their respective transformations, employing simple mathematical notation. From the leftmost to rightmost encoder (see Figure 2):

1. The slash represents location within the neural network. This encoder can be used to switch between layers
2. Scalar multiplication notation to symbolise applying a factor to all weights within the respective layer.
3. Δx to denote shifting all weights in the x-dimension, represented on the touch screen.
4. Δy to denote shifting all weights in the y-dimension, also represented on the touch screen.

Results

Output

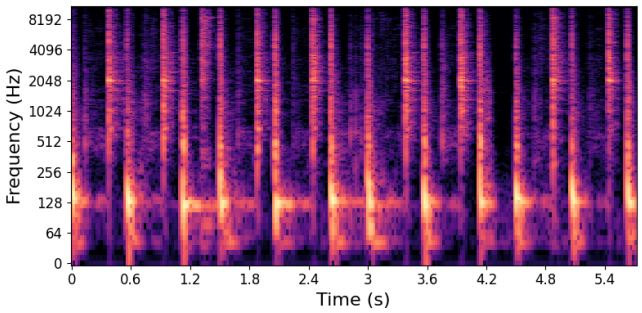
Through the user-feedback cycle and self-experimentation, network-bending transformations which led to "interesting" sonic results were identified. A tranformation example is shown as a spectrogram below, in Figure 1. The audio files and addtional examples are hosted at the following link: `https://danielmanz17.github.io/Brave-Samples/`. A longer instrument demo is presented in video format: `https://www.youtube.com/watch?v=0HugWkdesgw`.

[2] `https://github.com/danielmanz17/Brave`

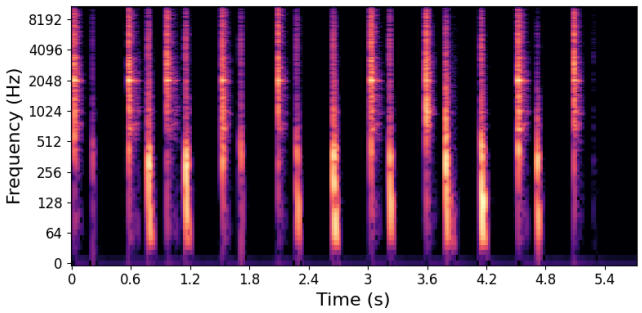
[3] `https://thepihut.com/products/iqaudio-dac`

[4] `https://learn.adafruit.com/adafruit-i2c-qt-rotary-encoder/overview`

[5] `https://www.waveshare.com/4.3inch-dsi-lcd.htm`



(a) Brave running the darbouka RAVE model without any transformations applied to the network architecture. Baseline for inspecting other outputs.



(b) Applying $\Delta y = 0.47$ at bias of first encoder layer. This transformation reduces the sustain of the percussive hits.

Figure 1: Spectrogram representation of audio before and after a network-bending transformation has been applied.

User-feedback

P1 noted that the encoders produced more predictable results than the touchscreen. For example, shifts in Δy applied to layer 17 seemed to consistently create a rhythmic "half step" transformation. They noted that the scale encoder could perhaps act as a "magnifier" - does increasing the scale of the tensor values at this layer amplify its corresponding processing fingerprint?

The need for an indication of location within the neural network was highlighted, as they felt "lost" within the model. To save sounds found within the parameter space, a preset functionality was suggested. High-dimensionality layers bear visual resemblance to a waveform which created confusion. Finally, it was noted that the encoder transformations are applied to the initial rather than the modified weights of the model.

P2 highlighted the need for continuous weight modification, indication of neural network location and visual differentation from waveforms. The Δy encoder steps were discrete rather than continuous, which didn't make sense to the user. In addition, the mapping direction of Δy transformation felt "unintuitive".

P3 noted that the lower dimensional layers were easier to control, and that perhaps these could be prioritised in layer exposure. They adopted a more rigorous approach to parameter space exploration - exploring specific transformations at each layer, and probing how this affects the audio output. For example, they investigated how specific shapes drawn on the touch screen at different layers would influence the output. Again, the issues of neural network location, preset saving and continuous weight modification were raised. In addition, an interpolation feature was suggested between the initial/modified layer, similar to a typical synthesiser dry/wet knob.

P4 enjoyed pushing the sounds as far as possible. They identified unexpected metallic/string like artefacts with certain bending approaches. Preset functionality and waveform confusion were highlighted.

P5 noticed that changing the values within the lower-dimensional layers seemed to generate more contained results. They found Δx produced rhythmic rather than timbral variation. The participant flagged confusion about the magnitude of the values - does the middle of the screen correspond to a null value of weights? In terms of the fabrication process, ribs/support pillars were suggested as the body felt fragile. Finally, higher encoder sensitivity was suggested.

P6 particularly enjoyed exploring the scale/bias encoders. They mentioned a soft learning curve - they were quickly able to determine what everything does. They found rhythmic divergence to be the most interesting line of exploration. Visual indication of location in neural network, input sample choice and waveform confusion were flagged. They mentioned some of the encoder transformations felt a bit "steppy".

These user feedback sessions led to the finalisation of the Brave instrument. This process of reflection and iteration will be discussed in the User-Feedback Reflection section. The final instrument is shown in Figure 2 below.

Discussion

Design and Development Process

The instrument could benefit from additional user-testing cycles, integrating quantitative data and more traditional HCI methodology (usability testing, think-aloud protocol, task analysis and A/B testing).

In addition, Brave could profit from a more diverse design methodology - introducing elements of the autobiographical framework (Ó Néill and Ortiz 2024). Brave could be integrated into a music production workflow/live performance setup, including experimentation with alternative RAVE models. Perhaps custom network-bending models could be trained - curating training corpora with network-bending as the end goal in mind.

It could be beneficial to consider alternative embedded systems - Jetson Nano7/STM32 32-bit Arm Cortex MCU8 would both be ideal candidates. These implementations could engage with pertinent questions regarding the future of embedded neural systems, enabling modification of the underlying architecture. Can we move towards a more scalable neural instrument? Could this evolve into a viable, consumer-facing product? Further miniaturisation and advances in neural processing architecture (compression) draw these questions ever closer.

User-Feedback Reflection

The absence of preset saving functionality was frequently flagged. This could be implemented with rectangular pads, each of which correspond to a saved network-bending preset. For example, holding the pad could save a new preset, and a singular press can recall the saved preset.

Another consistent pain point was getting "lost" in the neural network - the interface provides no indication for the user to triangulate their position within the network. This could be exposed via a number or layer name. A more advanced solution would be a graphical indicator using a touch-strip, which would provide a visual representation and control mechanism of layer position.

The discontinuous weight modification was frequently flagged - the encoder transformations are applied to the initial rather than the modified weights of the model, interrupting the user workflow.The authors hope to implement this in the future.

A few users commented that the sensitivity of the encoders could be increased, which would also reduce the perceived "stepiness" of the knobs. Sensitivity was increased by a factor of 1.4 for all transformation encoders. Further user-testing could be conducted to fine tune this. In addition, the momentary switch button was incorrectly detecting single/double taps for some users. The decision was made to use the push button functionality of the first encoder. This made intuitive sense as encoder one already controls layer position.

Finally, high-dimensional layers visually resemble a waveform. This created frequent confusion, especially for those with a background in electronic music production. Different UI approaches could be considered to expose the

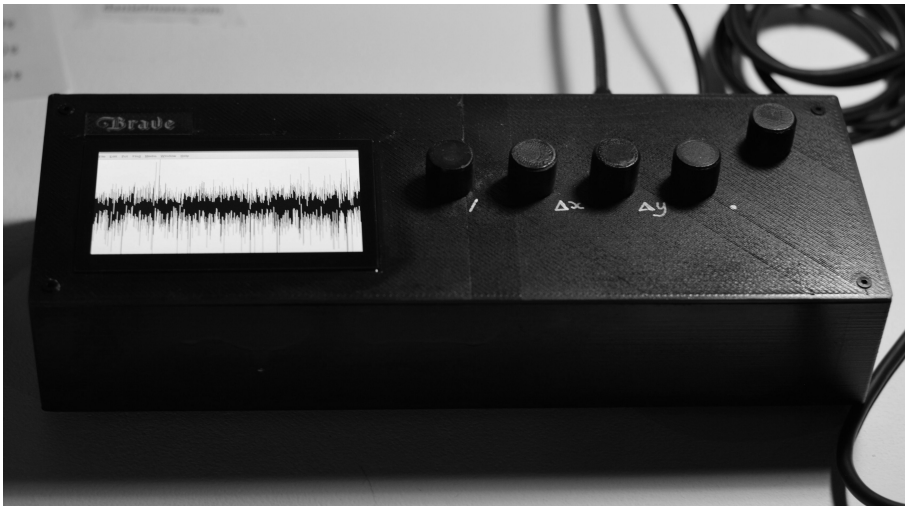


Figure 2: Final version of Brave.

layer weights - perhaps more traditional neural network representations accompanied by an activation heat map, switching the orientation of the screen, or reducing the dimensionality of the layer representations within the UI.

Conclusion

This study documents the design and fabrication of Brave: an embedded network-bending, stand-alone instrument. As neural audio synthesis becomes more widely used, there are growing concerns that its reliance on statistical averages could lead to the homogenisation of the cultural landscape, reinforcing a cycle of self-referential synthesis. This work seeks to explore new methods for addressing these challenges by contributing to the developing field of "network-bending". Incorporating musician-led feedback, user-centred interface development and accessible hardware integration, this study aims to lower technical barriers and foster broader participation within the network-bending framework.

References

Broad, T.; Leymarie, F. F.; and Grierson, M. 2021. Network bending: Expressive manipulation of deep generative models. In *Artificial Intelligence in Music, Sound, Art and Design: 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings*, 20–36. Berlin, Heidelberg: Springer-Verlag.

Bryan-Kinns, N., and Reed, C. N. 2023. A guide to evaluating the experience of media and arts technology.

Caillon, A., and Esling, P. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR* abs/2111.05011.

Fefferman, C.; Mitter, S.; and Narayanan, H. 2013. Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29.

Ghazala, R. 2004. The folk music of chance electronics: Circuit-bending the modern coconut. *Leonardo Music Journal* 14:97–104.

Haque, K. N.; Rana, R.; and Schuller, B. W. 2020. High-fidelity audio generation and representation learning with guided adversarial autoencoder. *IEEE Access* 8:223509–223528.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Huang, H.; Li, Z.; He, R.; Sun, Z.; and Tan, T. 2018. Introvae: introspective variational autoencoders for photographic image synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 52–63. Red Hook, NY, USA: Curran Associates Inc.

Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2021. Alias-free generative adversarial networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc.

Kingma, D. P., and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Ng, A., and Jordan, M. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T.; Becker, S.; and Ghahramani, Z., eds., *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Ó Néill, E., and Ortiz, M. 2024. From prototype to performance practice: Reflections on iterative instrument design. In *Proceedings of the 19th International Audio Mostly Conference: Explorations in Sonic Cultures*, AM '24, 439–444. New York, NY, USA: Association for Computing Machinery.

Semeniuta, S.; Severyn, A.; and Barth, E. 2017. A hybrid convolutional variational autoencoder for text generation. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 627–637. Copenhagen, Denmark: Association for Computational Linguistics.

Vahdat, A., and Kautz, J. 2020. Nvae: a deep hierarchical variational autoencoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. In *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, 125.

Vasquez, S., and Lewis, M. 2019. Melnet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083*.

Wang, W.; Gan, Z.; Xu, H.; Zhang, R.; Wang, G.; Shen, D.; Chen, C.; and Carin, L. 2019. Topic-guided variational autoencoder for text generation. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 166–177. Minneapolis, Minnesota: Association for Computational Linguistics.