# Network Bending as Circuit-Bending Inspired Live Neural Synthesis Hacking

Błażej Kotowski
blazej.kotowski@upf.edu
Universitat Pompeu Fabra
Barcelona, Spain

Frederic Font
frederic.font@upf.edu
Universitat Pompeu Fabra
Barcelona, Spain

## Abstract

Circuit bending has long framed technical malfunction and misuse as sites of musical invention, enabling non-expert, hands-on engagement with electronic sound circuits through the practice of an immediate canvas. This paper articulates *Network Bending* as a circuit-bending–inspired approach to neural audio synthesis, making its connection to the tradition and ethos of circuit bending explicit. Network Bending is presented here as realtime, hands-on modification of neural network parameters—specifically weights and biases—during sound generation, where instability, breakdown, and opacity become compositional material.

The paper situates this practice within histories of musical hacking and instrument repurposing, and introduces an open-source Network Bending tool that enables parameter-level intervention in neural audio models within a realtime environment. Drawing on repeated use in performances and workshops, we introduce the *Bending Log* as a compositional technique for documenting and reusing exploratory interventions across models and performance contexts, and present a complementary *Network Blending* experiment that combines corresponding parameter spaces across models. We further reflect on workshop-based dissemination and community uptake, positioning Network Bending as an instrument-centered performance practice, a pedagogical tool, and an exploratory engineering approach for engaging with the internal dynamics of neural audio systems.

## Keywords

network bending, neural audio synthesis, circuit bending, sound hacking, live performance

## 1 Introduction

Circuit bending is a historically established practice within experimental music, emerging in the late-twentieth century, which involves deliberately repurposing electronic sound-producing circuits beyond their intended design [12]. By intervening directly in existing circuitry, practitioners induce unexpected behaviors and sonic outcomes, treating technical failure as a raw composition material [5]. Crucially, circuit bending enables a critical engagement with technological systems without requiring prior expertise in electronics. Its practice is grounded in what is described as an *immediate canvas*, a mode of working in which alterations are performed hands-on, experimentally, and responsively, rather than through a comprehensive prior understanding of the system's internal logic [12].


*NIME '26, June 23–26, 2026, London, UK*


In this paper, we articulate *Network Bending* as a circuit bending inspired approach to neural audio synthesis. Although the term has originally been used by Broad et al., their formulation focuses on modifying intermediate activations during inference [2]. Our contribution extends Network Bending through the practice and ethos of circuit bending, shifting the site of intervention from activation space to the model's parameters. This distinction is clarified in Section 3.1.

In this context, Network Bending refers to exploratory, realtime interventions applied to the parameters of an audio neural network within a live setting. These interventions expose and repurpose instabilities and breakdowns in the model's behavior as material for musical composition, aligning the practice with traditions of hardware misuse, repair, and repurposing in experimental art [13, 17].

We have explored the concept through a series of live performances and disseminated it within the creative community via two in-person workshops, held respectively at a live coding conference and an open-source art festival. Through these activities, we experimented with various composition techniques, developed a dedicated interaction method, and articulated a set of practice-driven insights that inform both artistic and technical perspectives on the approach.

This article makes the following contributions:

- an articulation of Network Bending grounded in the ethos and practices of circuit bending
- an open-source implementation of a Network Bending tool for realtime neural audio intervention
- a reflection on Network Bending as a compositional technique, a political act, a pedagogical tool, and an exploratory engineering probe

A companion website[1] provides access to the instrument, including the Pd patch, the modified nn~ external, a video demonstration of its operation, and the tutorial video mentioned in sec 4.

The remainder of this article is structured as follows. In Section 2 we situate Network Bending within traditions of musical hacking and instrument repurposing, and outline properties of neural audio synthesis that position pretrained models as playable and modifiable sound instruments. In Section 3 we present the technique itself, detailing the technical implementation, introducing a graphical interface, proposing a compositional technique in the form of a *Bending Log*, and presenting the *Network Blending* experiment. In Section 4 we reflect on dissemination activities, including a series of workshops and a video tutorial, which led to community-developed projects built on top of the concept. In Section 5 we share insights and discuss limitations and future work. Finally, in Section 6 we share concluding remarks.

[1] https://network-bending-nime.github.io/

## 2 Background

This section provides background on two strands central to Network Bending: practices of hacking and repurposing in music, and the technical characteristics of neural audio synthesis that make such practices relevant in this context.

### 2.1 Hacking and Repurposing in Music

Technological reinterpretation has been of great importance to the development of electronic music throughout history [17]. From Cage's prepared piano and Schaeffer's experiments with tape, through Oliveros' use of test signal and noise generators, to experiments with electronic circuits by Ghazala and countless members of the circuit bending community around the world—musicians and composers have consistently sought creative misuse of technology.

While in the world of objects, makers have often engendered technological subversion through material modification, in the era of Digital Music Instruments (DMIs), hackers subvert their instruments through tinkering with code. Clearly, instrument innovation is not a domain exclusive to DIY cultures. In music technology, private companies and entrepreneurs have a strong interest in packaging technology into products in the form of proprietary, closed software. What separates hacking from this form of engineering are the social and political commitments of the practice.

The culture of hacking, rooted in the free software ethos, provides fertile ground for the emergence of communities of practice that explicitly push back against private copyrights, mass production, and other market-oriented logics—instead, fostering ideas of voluntary collaboration, mutual support, reuse of materials, and open source practices [17]. Additionally, as Levy documents in his book, participants often undertake projects "not solely to fulfill some constructive goal, but with some wild pleasure taken in mere involvement" [14].

Circuit bending serves here as a nominal example of DIY music hacking culture that embodies these political commitments and, as such, attracts artists motivated by its anti-establishment aspects. These commitments include the rejection of black-boxed and proprietary design, the reappropriation of obsolete or discarded technologies, and a hands-on mode of engagement that foregrounds misuse, repair, and modification as forms of technical and cultural agency, as elaborated by [13]. However, although Ghazala himself recognizes the inevitable political potential of the art form, he simultaneously emphasizes that what remains central to it is its capacity to "encourage a fresh musical thought" [12, p. 103].

Among the conceptual notions articulated by Ghazala, the idea of the *immediate canvas* is particularly significant in this work, as it encapsulates his deliberately anti-theoretical orientation toward electronic practice. The concept frames the electronic circuit not as an expert domain to be mastered through formal knowledge, but as a surface for direct, intuitive experimentation. Ghazala describes the concept as follows [12, p. 100]:

> The concept of an immediate canvas is very important. Until now, the assumed hurdles of electronic design have instilled a sense of apprehension in laypersons. Even if one does not electrocute oneself, a slow and tedious entry is expected via a stack of daunting, equation-bound texts. Circuit-bending changes all this, as it transforms the circuit into a friendly and "immediate" canvas like the painter's canvas: immediately there for anyone at all with brush in hand. Just walk up to it and paint. [...] The science of the electron is no longer needed to advance the creative moment. Finally, in electronics, one can just walk up and paint.

The ideas of creative hacking, bending, and recycling have been both explicitly and implicitly explored within the NIME community. Bendit_I/O, a project by Marasco et al., developed a low-cost hardware system enabling networked performance with circuit-bent instruments [16]. Concentric Sampler, by Tate, repurposes an obsolete floppy disk drive as an instrument for performance and sonic manipulation [20]. More recently, Evans et al. repurposed a pretrained machine learning model, originally trained to generate drum accompaniment, to drive a MIDI-enabled organ in an instrumental performance context [11]. In a related vein, Collins has explored code-level intervention in text-to-music generation systems as a form of creative destabilization [7]. A broader discussion of the ethos of repurposing within NIME is provided by Masu et al. [18].

Today, this ethos of reinterpretation extends to data-driven systems, as neural networks increasingly function as instruments within musical practice.

### 2.2 Neural Audio Synthesis

In this context, neural audio synthesis is an audio synthesis technique in which the synthesis process is driven by underlying neural networks [9]. Encompassing numerous methods with varying technical specificities, neural synthesizers share commonalities that can be attributed to the core characteristics of neural networks. A key characteristic is that computation is organized into a vast network of nodes arranged into layers, with connections described mathematically by weights and biases. These numerical values are derived gradually through a process of backpropagation, necessarily fueled by data. In the case of neural synthesizers, the network is typically fitted to a collection of sounds with the goal of controlled reproduction of the original sonic palette.

In contemporary practice, such systems have gained significant traction within creative communities. Yet the scale of modern models—often comprising tens of millions of trainable parameters distributed across dozens of layers—renders the meaning of individual stages of computation difficult to interpret, in contrast to more traditional synthesis techniques such as FM or additive synthesis. Although interpretability remains an active area of research, understanding the inner functioning of neural networks at a granular level often remains challenging. At the same time, the process of backpropagation is relatively slow and requires substantial computational resources, making the creation of neural synthesizers costly both economically and environmentally [19].

These characteristics—namely distributed parameterization, opacity, and high training cost—create conditions in which pretrained models become both difficult to fully interpret and attractive to reuse. In this sense, neural synthesizers present a form of internal circuitry whose complexity invites exploratory intervention and repurposing.

## 3 Network Bending

Building on the discussion of opacity, scale, and computational cost in contemporary neural synthesis presented in the previous section, we turn to Network Bending as a practical response to these conditions. Network Bending approaches pretrained models as material systems that can be directly intervened in, repurposed, and played, without requiring full interpretability or retraining. Drawing on the ethos of circuit bending and the notion of an immediate canvas, this section presents Network Bending as a situated technique for engaging with neural synthesizers through realtime parameter-level modification, enabling exploratory misuse and creative reuse in live and experimental contexts.

## 3.1 Terminological Clarification

The term *Network Bending* has previously been introduced by Broad et al. in the context of creative manipulation of generative models [2]. Within their taxonomy of active divergence strategies, Network Bending refers to modifications applied to intermediate activations during inference, “without making any changes to the weights” [1, p. 4].

Our approach differs at the level of intervention. Rather than manipulating activations within a fixed parameter configuration, we perform realtime interventions directly in the model’s parameter space—its weights and biases. Within Broad et al.’s taxonomy, such operations would align more closely with what is described as *Model Rewriting*.

We nevertheless retain the term *Network Bending* for two reasons. First, our work is explicitly grounded in the practice and ethos of circuit bending, emphasizing hands-on engagement with the internal structure of a system treated as a playable material substrate. Second, we treat the term as open and extensible, capable of encompassing both activation-level and parameter-level interventions under a shared logic of exploratory misuse. Through this framing, we contribute to an understanding of Network Bending as an instrument-centered practice grounded in hands-on engagement rather than a narrowly defined computational operation.

## 3.2 Technical Implementation

The Network Bending device is implemented through a Pure Data (Pd) frontend, built around a modified nn~ external that acts as an interface between the Pd environment and any compatible JIT-traced PyTorch model. Considering its open source nature, the readily available array of sound models, as well as its integration with Pd, we took RAVE as a point of departure for our experiment [4]. Later on, we also experimented with other compatible neural synthesizers, such as a custom model based on DDSP [10], previously developed by the first author.

*3.2.1 Modified nn~.* The system builds on nn~, a Pd external developed at IRCAM in the context of the RAVE project to enable realtime interaction with pretrained neural audio models[2]. We leverage this external for its convenience and existing support for neural audio synthesis. However, performing Network Bending requires direct access to the internal parameters of the model.

The original implementation of nn~ does not expose such access. To enable parameter-level intervention, we therefore forked and modified the external. Our modifications preserve its original functionality while extending it with an additional outlet and a set of bending-specific commands implemented at the existing command inlet.

These additions introduce three core commands: one for returning a list of all available network layers identified by name; a getter command for retrieving the full set of parameters associated with a specified layer; and a setter command for overwriting the parameters of a specified layer with a provided list of numerical values.

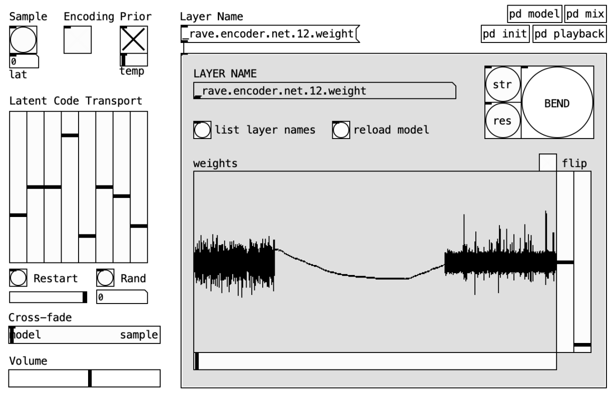


**Figure 1: Network Bending Pd interface.**

*3.2.2 Pd interface.* The central element of the Pd interface is a large display showing the parameters of the selected network layer. Layer selection is performed in two steps: first, the names of all available layers are printed to the Pd console; second, the selected layer name is copied into a dedicated message box at the top of the interface.

Once a layer is selected, its parameters can be modified in several ways. One may draw a new parameter shape directly with the cursor or apply predefined mathematical operations using dedicated sliders, including addition (offset), multiplication (scaling), and negation (flipping). An additional slider enables circular shifting of the parameter array. These operations can be combined in any order.

Applying a bending operation writes the modified parameter array back into the model through an nn~ operation, using the buttons positioned above the display. A temporary buffer can be accessed using the store and restore buttons, allowing intermediate states to be recalled. The original model parameters can be restored at any time by reloading the model.

In addition to the bending-specific controls, the left side of the interface provides standard nn~ functionality, including latent code manipulation, unconditional generation using a prior, and realtime style transfer of a selected audio file. A detailed description of these controls can be found in the RAVE and nn~ tutorial provided by IRCAM[3].

## 3.3 Composition Techniques

This subsection presents two compositional approaches used in our work with the Network Bending device. The first, the *Bending Log*, was developed through iterative experimentation with the system. The second, Model Blending, was incorporated into our implementation based on prior work.

*3.3.1 Bending Log.* Considering the nature of machine learning–derived sonic models, the variety of their architectures, as well as their source data, the computational graphs of the models can differ significantly on a case-by-case basis. For this reason, the bending of each model needs to be considered a separate

[2]https://github.com/acids-ircam/nn_tilde

[3]https://forum.ircam.fr/article/detail/tutoriel-rave-and-nn/

**Table 1: Fragment of Bending Log for a custom DDSP model.**

| Layer ID | Layer name | Operation | Result |
|---|---|---|---|
| 17 | pretrained.decoder.input_latent_bottleneck.0.bias | roll to the right | produce more harmonic and brighter timbres |
| 17 | pretrained.decoder.input_latent_bottleneck.0.bias | squeeze | sharper, more compressed sound (a bit like an OTT) |
| 24 | pretrained.decoder.input_latent_bottleneck.6.weight | roll to the right | smear the transients |
| 26 | pretrained.decoder.input_latent_bottleneck.7.weight | scale | low – lower volume, higher dynamics; mid – introducing distortion; high – introducing extra tonal timbral elements |
| | *... additional layers omitted ...* | | |
| 44 | pretrained.decoder.gru.weight_ih_l1 | scaling | contracting – fast rhythmic modulation |
| 44 | pretrained.decoder.gru.weight_ih_l1 | rolling | all kinds of timbral shifts |
| 52 | pretrained.decoder.inter_mlp.3.weight | rolling | small room reverbs and various distortions |
| 52 | pretrained.decoder.inter_mlp.3.weight | drawing | smearing and drone |
| 58 | pretrained.decoder.inter_mlp.7.weight | offseting | upwards – adding noise and reverb; downwards – spectral modeling style effects |

project. The Bending Log is an exploratory technique derived from the description of bending process described by Ghazala, enabling the discovery of compelling "circuit paths" through systematic shorting of the circuit board [12].

The Bending Log is constructed similarly, but in the digital domain. The composer iterates over individual layers of the model, introducing various changes in their parameters intuitively. The modifications are introduced in isolation, while listening to their results. Once an interesting sonic transformation is discovered, the operation is noted down in a table, saving the layer index and name, alongside the type of operation applied and a description of its sonic result. Optionally, logged operations from this initial pass can then be combined with one another and the resulting composite transformations added to the log. In this way, a rich collection of distinct sound transformations is discovered, which can later be used in live performance or a composition process. Table 1 exemplifies a log derived from work on a custom DDSP model.

*3.3.2 Model Blending.* Although not planned at the outset of the project, the second technique arose from the affordances of the modified nn~ external. During experimentation, the ability to retrieve and overwrite entire parameter arrays made it feasible to combine corresponding layers from different models. This procedure draws inspiration from earlier layer-swapping experiments by Collins et al. and corresponds to what Broad et al. termed *Network Blending* [1, 8].

In contrast to the Bending Log, which relies on intuitive, layer-specific manipulation, Model Blending combines parameters from two distinct models through controlled interpolation. The technique is implemented by extending the Pd patch with buffers for temporarily storing layer parameters (see fig. 2). To perform a blend, the composer first loads model $A$ and stores the parameters of a selected layer in buffer $X$. Model $B$ is then loaded, and the parameters of the corresponding layer are stored in buffer $Y$. A new parameter set is computed through linear interpolation,

$$Z = (1 - k)X + kY, \quad k \in [0, 1]$$

where $k$ is an interpolation factor that controls the contribution of each model. The resulting array $Z$ is then written back into the model.

For the operation to function correctly, the corresponding layers in both models must have identical dimensionality. In practice, blending is therefore most viable between models that share the same architecture but have been trained on different sound archives.

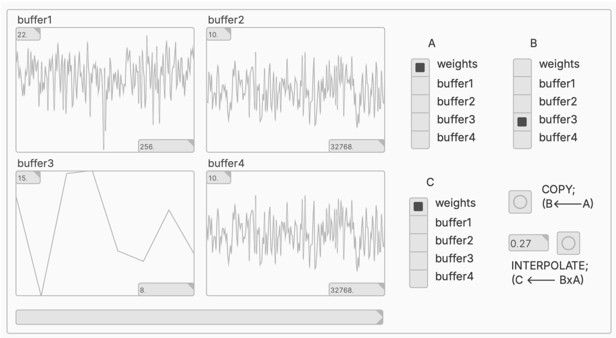


**Figure 2: Model blending extension featuring multiple parameter buffers.**

## 4 Community Dissemination

To evaluate Network Bending as a performance technique and to engage it within a broader creative community, the project was disseminated through live performances, workshops, and online materials. Two in-person workshops were held, one in the context of a live coding conference and one at an open source art festival. In addition, in response to community requests, a video tutorial was published online, introducing the concept and guiding users through installation and basic use of the tool.

### 4.1 Workshops

We organized two in-person workshops: one in the context of a conference and one at an open source art festival. Each workshop involved approximately twenty participants. One session lasted four and a half hours, while the second was limited to two hours. Both workshops followed a similar structure. We began with a theoretical introduction covering histories of instrument hacking and a brief overview of neural networks and neural network–based sound practices. This was followed by a practical session in which participants installed the modified Pd external and experimented with the Network Bending patch. In the

**Figure 3: Workshop on Network Bending at Art Meeds Radical Opennes 2024 (AMRO24).**

longer workshop, participants also trained small nn∼-compatible DDSP-based models on compact sound collections.

The workshops proved effective in demystifying neural synthesis and enabling direct engagement with neural networks. Participants engaged in critical discussion, hands-on experimentation, and exchange of ideas. As expected in a technically demanding setup, installation posed challenges for some participants, particularly those with less technical experience. In several cases, participants shared systems when installation could not be completed within the available time, suggesting that more streamlined, cross-platform support would further improve accessibility. These constraints were more visible in the shorter workshop, where the limited duration reduced the time available for experimentation.

Based on these experiences, future iterations of the workshops could reduce the amount of theoretical material and move more quickly toward hands-on interaction, aligning more closely with the notion of an immediate canvas. Nevertheless, all materials were shared with participants, and some continued to engage with the project afterward by requesting clarification and guidance, indicating sustained interest beyond the workshop setting.

### 4.2 Community Projects

**Figure 4: Brave: A dedicated hardware interface for the Network Bending patch built by Manz et al. [15].**

The workshops, together with the release of documentation and the video tutorial, led to several voluntary, community-driven contributions. One contribution took the form of a Pd external that reimplemented and extended the original patch with an improved internal structure, enhancing usability and accessibility[4]. Another outcome was a graduate thesis project that developed a dedicated hardware interface for interacting with the Network Bending system, featuring physical controls mapped to the implemented parameter operations [15] (See fig. 4). In addition, one participant created a Max/MSP version of the tool and contributed a corresponding Max nn∼ object to the repository.

These developments indicate that disseminating Network Bending fostered continued experimentation and technical extension within an open source, practice-oriented environment.

## 5 Discussion

Network Bending proved effective as a pedagogical tool, particularly in its capacity to demystify neural networks and open experimentation to a broader audience beyond technically specialized practitioners. By exposing internal network parameters through an intuitive interface, the technique enabled hands-on engagement with neural synthesis systems and lowered barriers to participation. In this sense, Network Bending demonstrates how hacking and open-source practices can function as political acts, redirecting engineering efforts toward alternative goals such as pedagogy, community building, and creative exploration.

At the same time, the technique generated a wide range of compelling sonic results. Direct manipulation of network weights and biases produced glitches, abrupt shifts, unstable textures, and timbres that diverged significantly from the original capacities of the bent models. Many operations, of course, produce silence, noise, or divergent output without musical interest; the productive glitches in the Bending Log are a filtered subset surfaced through listening. These behaviors became primary compositional material rather than errors to be corrected. The instability of the system was embraced as a source of expressive variation and surprise.

Simply providing direct, interactive access to neural network parameters fostered opportunities for critical discussion within the community. This openness supported collective experimentation and exchange, which in turn led to the development of creative and exploratory projects building on the original technique. Rather than positioning neural synthesis as a closed or expert-only domain, Network Bending encouraged shared inquiry and situated practice, allowing participants to engage with neural networks as mutable and negotiable systems. This orientation resonates with ongoing discussions in NIME on the longevity and reactivation of deep generative models [6] and with broader efforts to demystify AI through explainable and artistically grounded practice [3].

One limitation observed empirically, and also reported through an extended user study in related work, concerns the lack of support for saving or exporting modified, or “bent,” network states [15]. As in circuit bending practice, where exploratory interventions are often stabilized into self-contained instruments through hardwiring, the ability to preserve and reuse bent neural networks could significantly extend the utility of the technique. Future work could therefore explore mechanisms for fixing modified network parameters into persistent instruments with dedicated controls that can be reloaded and performed.

[4]https://martstil.de/code/neural-network-bending-in-pure-data

Interestingly, beyond artistic and pedagogical contexts, Network Bending also exhibits latent potential as a tool aligned with more conventional engineering objectives. Through live interaction with network weights and biases, the technique affords a form of informal, exploratory ablation, enabling the systematic investigation of the role and behavior of specific layers during sound generation. This suggests possible applications in probing and understanding neural synthesis models beyond purely creative use.

Finally, a more systematic study of Network Bending could be undertaken, including experimentation with more complex or intentional transformations of layer weights. Such work could help clarify the relationship between specific parameter interventions and resulting sonic behavior, further situating Network Bending as both an exploratory artistic practice and a reflective engineering method.

## 6 Conslusions

This paper introduced Network Bending as a circuit-bending inspired approach to working with neural audio synthesis systems in realtime. It described an open source implementation that allows for direct, intuitive intervention in neural network parameters during sound generation, as well as practice derived composition techniques supporting exploratory and repeatable performance. The paper further reflected on the use of Network Bending in performance, workshop, and open source contexts, considering its role as a pedagogical entry point, a community driven practice, and an exploratory engineering approach to neural sound systems.

## 7 Ethical Standards

The authors used ChatGPT for language editing to improve grammar and clarity. All scientific content, interpretations, and conclusions were developed by the authors.


## Acknowledgments

This work has been supported by (1) IA y Música: Cátedra en Inteligencia Artificial y Música (TSI-100929-2023-1), funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial and the European Union-Next Generation EU, and IMPA: Multimodal AI for Audio Processing (PID2023-152250OB-I00), funded by the Ministry of Science, Innovation and Universities of the Spanish Government, the Agencia Estatal de Investigación (AEI) and co-financed by the European Union, (2) the Maria de Maeztu Strategic Research Program (CEX2021-001195-M) and (3) the IMPA Project PID2023-152250OB-I00 funded by MCIU/AEI/10.13039/501100011033/FEDER, UE.


## References


[1] Terence Broad, Sebastian Berns, Simon Colton, and Mick Grierson. 2021. Active Divergence with Generative Deep Learning - A Survey and Taxonomy. In *ICCC*. Association for Computational Creativity (ACC), 227–236.

[2] Terence Broad, Frederic Fol Leymarie, and Mick Grierson. 2022. Network Bending: Expressive Manipulation of Generative Models in Multiple Domains. *Entropy* 24, 1 (2022). https://doi.org/10.3390/e24010028

[3] Nick Bryan-Kinns, Shuoyang Zheng, Francisco Castro, Makayla Lewis, Jia-Rey Chang, Gabriel Vigliensoni, Terence Broad, Michael Paul Clemens, and Elizabeth Wilson. 2025. XAIxArts Manifesto: Explainable AI for the Arts. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA 2025, Yokohama, Japan, 26 April 2025- 1 May 2025*, Naomi Yamashita, Vanessa Evers, Koji Yatani, and Sharon Xianghua Ding (Eds.). ACM, 650:1–650:8. https://doi.org/10.1145/3706599.3716227

[4] Antoine Caillon and Philippe Esling. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR* abs/2111.05011 (2021). arXiv:2111.05011 https://arxiv.org/abs/2111.05011

[5] Kim Cascone. 2000. The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music. *Computer Music Journal* 24, 4 (2000), 12–18. http://www.jstor.org/stable/3681551

[6] Isaac Clarke, Francesco Ardan Dal Rí, and Raul Masu. 2025. Longevity of Deep Generative Models in NIME: Challenges and Practices for Reactivation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 224–230.

[7] Nick Collins. 2025. Unstable Audio: Code Bending Text-to-Music Generation. In *AES International Conference on Machine Learning and Artificial Intelligence for Audio*. London.

[8] Nick Collins, Vít Ruzicka, and Mick Grierson. 2020. Remixing AIs: mind swaps, hybrainity, and splicing musical models. In *AIMC*.

[9] Mateo Cámara, Fernando Marcos, Anders R. Bargum, Cumhur Erkut, Joshua Reiss, and José Luis Blanco. 2026. Neural Audio Synthesis for Sound Effects: A Scope Review. *IEEE Transactions on Audio, Speech and Language Processing* 34 (2026), 427–445. https://doi.org/10.1109/TASLPRO.2025.3646080

[10] Jesse H. Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. 2020. DDSP: Differentiable Digital Signal Processing. In *ICLR*. OpenReview.net.

[11] Nicholas Evans, Behzad Haki, and Sergi Jordà. 2025. Repurposing a Rhythm Accompaniment System for Pipe Organ Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Zenodo, 116–120. https://doi.org/10.5281/zenodo.15698807

[12] Qubais Reed Ghazala. 2004. The Folk Music of Chance Electronics: Circuit-Bending the Modern Coconut. *Leonardo Music Journal* 14 (12 2004), 97–104. https://doi.org/10.1162/0961121043067271 arXiv:https://direct.mit.edu/lmj/article-pdf/doi/10.1162/0961121043067271/1674123/0961121043067271.pdf

[13] Garnet Hertz and Jussi Parikka. 2012. Zombie Media: Circuit Bending Media Archaeology into an Art Method. *Leonardo* 45, 5 (10 2012), 424–430. https://doi.org/10.1162/LEON_a_00438 arXiv:https://direct.mit.edu/leon/article-pdf/45/5/424/1575092/leon$_{a0}0438.pdf$

[14] Steven Levy. 2001. *Hackers: heroes of the computer revolution* (updated ed.). Penguin, London. 464 pages.

[15] Daniel Manz, Mick Grierson, et al. 2025. Brave: Designing an Embedded Network-Bending Instrument, Manifesting Output Diversity in Neural Audio Systems. In *ICCC*. Association for Computational Creativity (ACC).

[16] Anthony T Marasco, Edgar Berdahl, and Jesse Allison. 2019. Bendit_I/O: A System for Networked Performance of Circuit-Bent Devices. In *Proceedings of the International Conference on New Interfaces for Musical Expression*.

[17] Raul Masu and Fabio Morreale. 2021. Composing by hacking: Technology appropriation as a pedagogical tool for electronic music. In *Teaching Electronic Music*. Routledge, 157–171.

[18] Raul Masu, Fabio Morreale, and Alexander Refsum Jensenius. 2023. The O in NIME: Reflecting on the Importance of Reusing and Repurposing Old Musical Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Zenodo, 106–115. https://doi.org/10.5281/zenodo.11189120

[19] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *ACL (1)*. Association for Computational Linguistics, 3645–3650.

[20] Timothy Tate. 2022. The Concentric Sampler: A Musical Instrument from a Repurposed Floppy Disk Drive. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. https://doi.org/10.21428/92fbeb44.324729a3