



Audio Engineering Society Conference Paper 15

Presented at the AES International Conference on Machine
Learning and Artificial Intelligence for Audio
2025 September 8–10, London, UK

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (<http://www.aes.org/e-lib>), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Unstable Audio: Code Bending Text-to-Music Generation

Nick Collins¹

¹*Durham University Music Department*

Correspondence should be addressed to Nick Collins (nick.collins@durham.ac.uk)

ABSTRACT

Text-to-music models provide new audio generational and transformational capability, though their control is not a precise art. Embracing the aesthetics of ‘code bending’, glitch musicians’ approach to software manipulation and the sounds of noise music, the open source Stable Audio 1.0 model is manipulated to enable radical new audio outputs. The domain of historical late 1980s Detroit techno is used as a constrained arena for generation, based on a fixed text prompt. A comparative study is then undertaken of perturbing model parameters for each of the 745 layers of the model. Adding random noise to layer parameters is more likely to lead to interesting results than zeroing (pruning) connections or randomly reconfiguring parameters throughout the model, though all manipulations had some chance of creating interesting outputs. Further potential code bending projects for text-to-music engines are discussed.

1 Introduction

Deep learning has established an immensely productive and active research area for sound synthesis, in neural audio synthesis. For instance, the original WaveNet model [1] has had over 6000 citations, and subsequent developments have included large dataset trained company efforts such as Meta’s AudioCraft¹ with open source code and freely downloadable models [2], and multiple audio models available through pipelines in the Hugging Face transformers² and diffusers³ python packages. Such publicly available software is an irresistible target for subversion in artistic projects, and neural audio engines are powerful sound synthesis resources which may be approached with more experimental aesthetics.

The area of code bending [3] is a development which crosses hardware hacking [4] and live coding [5, 6], promoting the deliberate rewriting and perturbation of existing software for novel aesthetic ends. In the area of deep neural networks, this has been called ‘network bending’ [7] and is applicable to generative models for the audio, visual and joint audiovisual modalities [8]. In Collins et al. [9] open ‘brAIIn’ transplant surgery is explored swapping layers between multiple autoencoder models; Dzwonczyk et al. [10] bend the Stable Diffusion image generation engine to create music video outputs, by using audio features as control parameters to manipulate nodes to render a sequence of multiple images. The corruption of neural audio models through such techniques can lead to new timbres and new control possibilities, empowering new compositions utilising these sounds, and providing means to escape the more obvious close reproductive intent of standard text-

¹<https://github.com/facebookresearch/audiocraft>

²<https://huggingface.co/docs/transformers/en/index>

³<https://huggingface.co/docs/diffusers/index>

to-audio models. Such a project steers closer to glitch [11] and noise music [12] aesthetics than conventional computer science oriented deep learning work.

In this paper we explore hacking Stable Audio Open (1.0) which is an open source text-to-music model. It utilises a DiT (Diffusion-Transformer) model on the inner representational (latent) space of an associated audio encoder-decoder. Technical details of Stable Audio 1.0 are discussed in reports from Stability-AI researchers [13, 14]; the model with all source code and weights is available online.⁴ This paper carries out a comparative investigation of manipulating the parameters at the many different layers of the Stable Audio generative model in section 2. We discuss the sound transformations and the potential for further code bending in section 3. Sound examples⁵ and open source code⁶ for the manipulations and figure generation accompany the paper.

2 Manipulations

Stable Audio 1.0 has around 1.2 billion float parameters, with 745 layers in the core model combining the autoencoder which accesses the lower dimensional latent space where generation can take place, a diffusion-transformer model for longer-structure aware generation, and text embedding module for conditioning generation on text prompts. The number of parameters available at each layer vary but can become three dimensional tensors with millions of parameters (for example, layer 570 labelled *pretransform.model.decoder.layers.1.layers.1.weight_v* has shape [2048, 1024, 16]). In section 2.1 we explore adding Gaussian noise of different amplitudes to the set of parameters at particular layers. In section 2.2 a subset of parameters across the entire model are either zeroed out, or set to a random value; this latter manipulation leads to more unstable outputs, including silent output.

2.1 Adding noise to model layers

In order to better quantify the effect of interventions on the many potential hacking points within a large model, an exhaustive search was carried out across all the 745 layers of the Stable Audio model. To each

tensor of parameters for a given layer, Gaussian noise was added⁷ of amplitude 0.03, 0.3 and 3 (in initial experiments, 0.01 was too little to hear much difference in many cases). Gaussian noise was selected as a baseline stochastic model rather than a uniform distribution, but given the order of magnitude differences here, the type of noise was not particularly critical. A fixed seed and fixed prompt was used so that the baseline generation would always be the same and only the change in weights would affect the output; outputs were all 10 seconds long, at 44.1KHz. The prompt was lifted from the history of electronic dance music, namely Derrick May's description of Detroit Techno: 'George Clinton and Kraftwerk are stuck in an elevator with only a sequencer to keep them company' [15]. This prompt was chosen because of the author's familiarity with the history of electronic dance music in this era and electronic music history more generally [16].

Rendering of three added noise variants for each of 745 layers took 31 hours on a recent MacBook Pro with Apple M4 Pro chip (utilising the Metal Performance Shader GPU). Once obtained, the 10 second outputs from Stable Audio were analyzed with respect to three timbral features (spectral centroid, spectral bandwidth and spectral flatness) extracted with librosa⁸ [17] and averaged over the 10 second clips. To provide further context to the text-to-music generation, a small corpus of control cases was also introduced, and analyzed with the same three audio features. The additional audio sources consisted of 10 second extracts from Kraftwerk (15 famous tracks from 1975 to 1981), Parliament Funkadelic (eight tracks from the *Mother-ship Connection* album released in 1975), and Derrick May (10 well known tracks from 1985-1989, sourced from his compilation *Innovator* (1997)). Further, 10 variations (without weight manipulation and without any set seeds) were also created with Stable Audio 1.0 for each of the five prompts below, including the original prompt above and some related descriptions of the historical style:

- 'Detroit techno circa 1988'
- 'Derrick May electronic dance music'
- 'early techno music'

⁴<https://github.com/Stability-AI/stable-audio-tools>

⁵<https://composerprogrammer.com/unstableaudio.html>

⁶<https://github.com/sicklincoln/UnstableAudio>

⁷Using `torch.randn`, which gives a normal distribution of mean 0 and standard deviation 1, scaled by the coefficients here.

⁸<https://librosa.org/doc/latest/index.html>

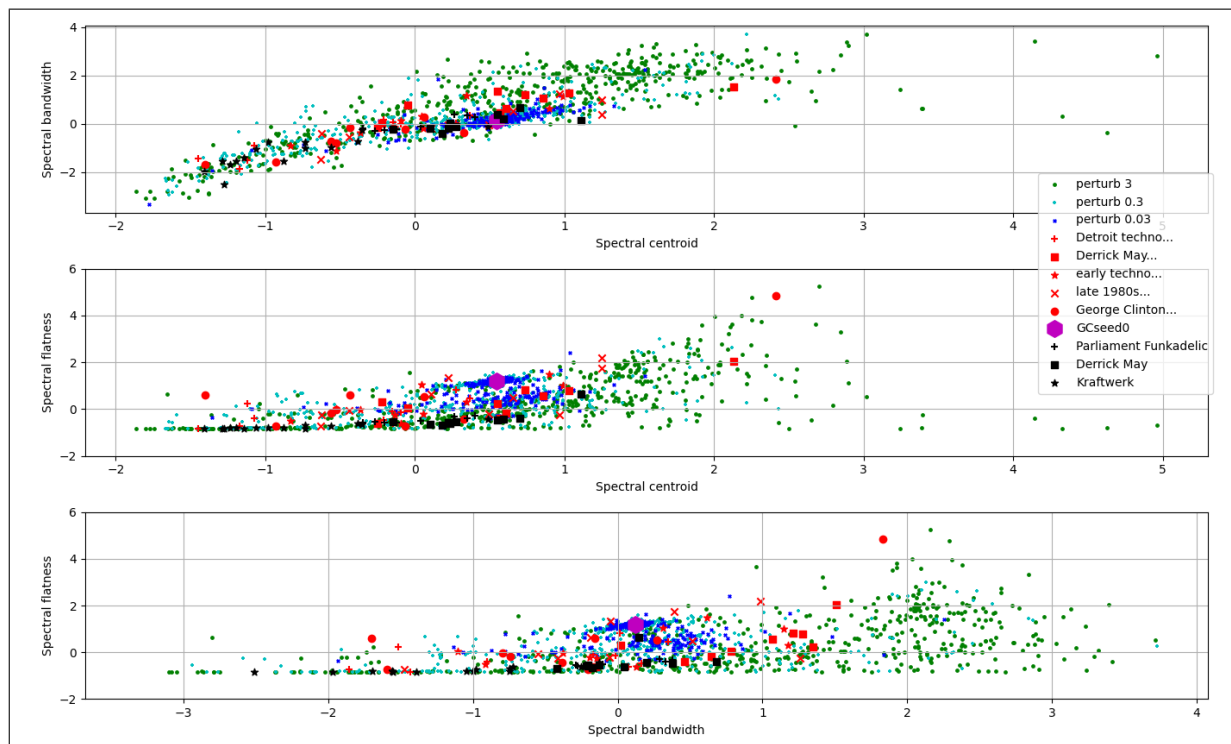


Fig. 1: Three scatter plots showing the interactions of three timbral features, plotted for Stable Audio generations with different prompts, with different layer parameter manipulations, and with ground truth historical audio from Kraftwerk, Parliament Funkadelic and Derrick May.

- ‘late 1980s Detroit techno’
- ‘George Clinton and Kraftwerk are stuck in an elevator with only a sequencer to keep them company’

Figure 1 plots these different conditions with respect to each other. The original Stable Audio generation based on the ‘George Clinton’ prompt with seed 0 and no weight variation is indicated as a larger point in purple. Figure 3 extends the analysis to rhythmic features, extracted using the SuperCollider Music Information Retrieval library [18]. The two features in the plot are the mean IOI (Inter-Onset Interval) as found by a percussive onset detector, and the ‘metricity’ which is a measure of how metrically aligned the beat histogram is in terms of simple integer harmonic or sub-harmonic relationships of peak locations to the largest peak. The historic human-composed music is generally separated especially on the metricity dimension (for convenience the figure also include audio examples of the overall zeroing and noise conditions of section 2.2).

The outputs from the Detroit techno prompts, from a musicological standpoint, are highly unreliable. Not only do they include anachronistic sounds, but they are often not electronic dance music at all but screaming walls of noise. The weight manipulated outputs did lead to interesting sonic variations, including the introduction of voice-like sounds (despite Stable Audio not being trained for vocal reproduction) and ‘ambient’ transformations from the original more noisy rhythmic material, to spaced out ‘elevator music’ far from techno’s pulsing drive.

The three features used here are only a rough timbral guide to the music and do not include rhythmic or pitch information, or time varying ‘spectromorphology’ [19]. Nonetheless, they show that the prompt based generations, and the parameter manipulations of the model, swarm around but do not often come close to the original Kraftwerk, p-funk and Derrick May works, and more often diverge into unrelated territory. Human aural analysis confirms this.

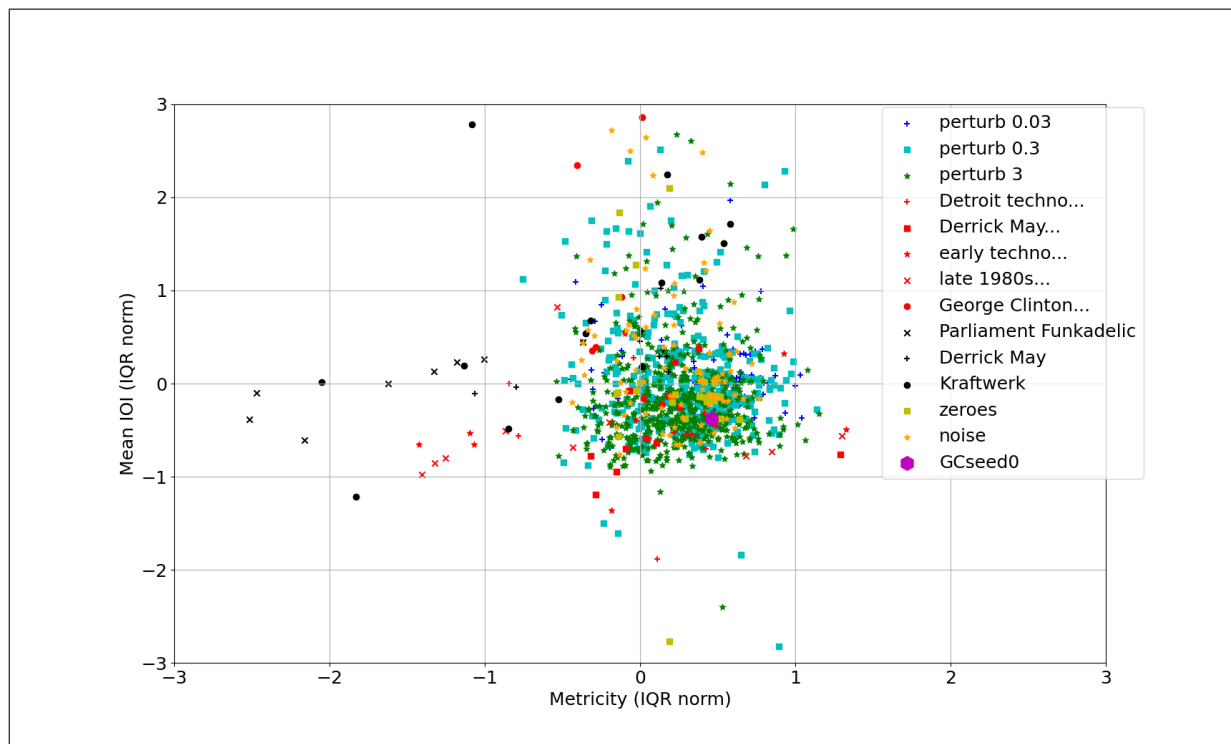


Fig. 2: Plot of two rhythmic features (mean IOI and metricity), plotted for Stable Audio generations with different prompts, with different layer parameter manipulations, and with ground truth historical audio from Kraftwerk, Parliament Funkadelic and Derrick May.

Interventions have different effect based not only on what amplitude of Gaussian noise is added to a layer’s parameters, but on the stage of the layer through the model. Figure 3 plots the three features against layer. It is initially striking that a run of layers, counts 380 to 562, do not adjust the output at all, seen in the feature data at a fixed plot level; zooming revealed no finer grained structure, and aural checks showed only the original seed 0 audio result was being returned. However, closer investigation of the model proved that this was not impactful in final rendering because it corresponds to the *pretransform.model.encoder* layers, which are only utilised in audio to audio transformation and not when going from a text prompt direct to output audio (adjusting weights for layer 563, the beginning of the following *pretransform.model.decoder*, shows audible results again). If starting from input audio, it would be interesting to manipulate the encoder weights, but wasn’t a task explored herein.

2.2 Zeroing out and randomising a subset of weights

Rather than adding a fixed tensor of perturbations, it is possible to directly change existing parameters, though this can have a more unsettling effect on the model.

Fixing the same prompt and seed as previously, a random set of parameters picked from any layer were set to zero, for ten different conditions: 1, 10, 30, 100, 300, 1000, 3000, 10000, 100000 and 1000000 zeroings. Each experiment was repeated 10 times (so across the 10 conditions, 100 runs of Stable Audio 1.0). The more parameters zeroed out, the more likely a silent output would be, though this only became a prominent issue once going past 1000 zeroes. Many outputs, especially for a smaller number of zeroes, had no perceptible difference to the reference starting point (seed 0, no zeroes, no parameter adjustments). However, a few more radical transformations were achieved for a moderate number of zeroes (300-3000). Since zeroing out parameters is equivalent to ‘pruning’ connections

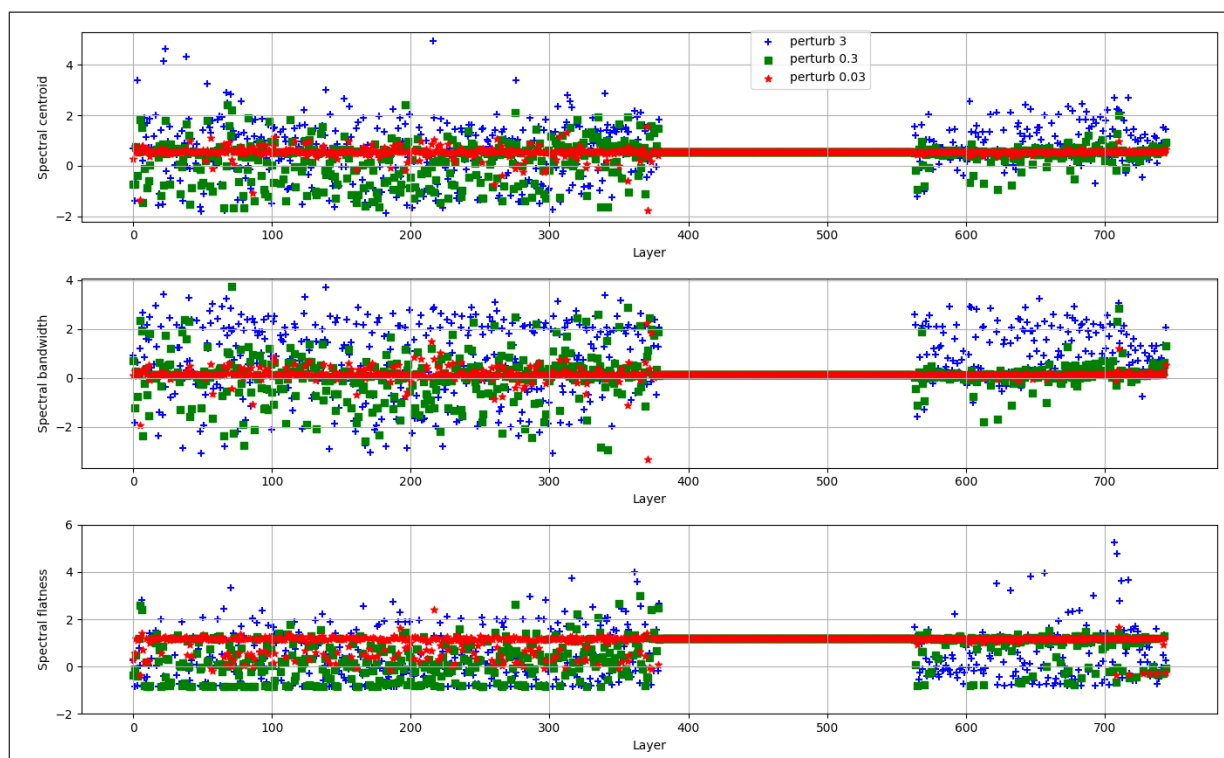


Fig. 3: Three scatter plots showing three timbral features plotted against the layer of network receiving a parameter adjustment.

in a network, the model was only robust to moderate pruning (around three in one million connections, 0.0003%).

We also investigated, again with the same prompt and seed as previously, setting a random set of parameters from any layer to a uniform random number from -1.0 to 1.0, for 1, 10, 30, 100, 300, 1000, 3000, 10000, 100000 and 1000000 parameter substitutions, each experiment repeated 10 times (another 100 total runs of Stable Audio 1.0). For higher numbers of randomised parameters (≥ 10000), outputs tended to become high frequency repetitions, sounding very coarse and not distinctive; nonetheless, when setting a moderate number of parameters (100, 300, 1000, 3000) it was possible to access new sounds; 1000-3000 zeros were especially good at transforming sound in surprising ways.

3 Discussion

Hacking the Stable Audio 1.0 model does lead to some productive and intriguing sounds. Audio examples

are released to accompany this paper; they range from frequency filtering effects, to more radical slurring, timbral substitution, stuttering, and changes of reverberant acoustic. Most outputs were continuous in rhythmic or ambient texture; a few demonstrated isolated hits with silence inbetween. Occasionally, spectral processing artefacts were obvious, but most had a nice sound quality, if sometimes distorting. Unless parameters were entirely zeroed out, often leading to silent output, most parameter changes would allow through a transformed version of the original output. The systematic comparison by layer above only went up to a parameter multiplier of 3; higher values still are perfectly possible, but in qualitative trials more often lead to extreme distortion, or ‘frozen’ outputs with static frequency partials or constant stridulation.

Musicians are actively exploring text-to-audio generative AI models [20], and more facilities for their customisation, personalisation and sonically impactful code bending warrant additional research. The manipulations investigated here are not undertaken to repro-

duce well worn paths in existing music (one of the limitations of general text-to-audio training founded in specific databases) but to push models outside of the more obvious places in their generative space, and access new sound transformations. With this goal in mind, the project did reveal many unexpected sound outputs that could not be reached in alternative obvious ways with existing computer music algorithms. A more systematic polling of layer by layer weight perturbations led to more interesting results overall than a global manipulation such as substituting an activation function with another; however, given that only limited text prompts have been systematically compared herein, there is much scope to compare manipulations in the case of other prompts and no hard rule can be established at this point.

Future techniques to be explored might include:

Coarse tuning It is possible to further train the baseline Stable Audio 1.0 model on a given audio corpus, via fine tuning of all model weights, or substantially reduced numbers of parameters via techniques like Low Rank Adapters (LORAs) [21, 22, 23, 24]. Rather than fine tune an existing model, we might ‘coarse tune’, maximising unexpected outputs by training on non-congruent audio, connecting an under-sized parasitical LORA to awkward layer positions, using a deliberately wild cost function etc.

Model cannibalization Varying the experiments of Collins et al. [9], modular blocks such as the encoder, or specific layers, can be substituted from alternative models, shuffled (where there are compatibilities of tensor shape, for instance, between down and up sections of the encoder-decoder), or additional layers added in to disrupt normal model behaviour.

Short circuits Following the lead of hardware hacking [4], additional rogue connections can be added between any artificial neurons in any layers of the network.

The slowness of Stable Audio inference does not facilitate deployment in live performance except perhaps via background rendering.

At the time of writing, a Stable Audio 2.0 model exists and can be tried online⁹, but has not been open

⁹<https://stableaudio.com/>

sourced. The five Detroit techno prompts used above were tested with SA2; plausible techno could be created, but were not convincing historically. Just as for Stable Audio 1.0 examined herein, this is not especially unsurprising, since the model was ethically trained on non-copyrighted audio, not including actual Detroit techno in the training corpus. Specialisation of the model towards historical accuracy could be undertaken, at the likely expensive of generality (as found for MusicLDM in Ronchini et al. [20]). Whether the range of sound transformation possible through model hacking would be more compromised for an over-fitted fine-tuned model is an open question. Research in this area depends on open source code availability; hacking of the suno or udio proprietary models, for example, is not practicable at this time.

The manipulations detailed in this paper are not immediately available to musicians to explore without use of the source code provided directly (though it doesn’t take many adjustments to hack a particular layer and change the prompt and noise amplitude). A more used friendly interface, and a more wide ranging user study across a free choice of prompts and model manipulations, would be a further interesting step.

4 Summary

This paper investigated sound transformations achievable by manipulating the parameters of the Stable Audio 1.0 model. Keeping generation fixed to a set random number seed and prompt, parameter changes within and across layers were investigated, often leading to interesting sound transformations. The model was able to access sounds well outside of the norm, not by reaching new places in the innermost latent space of the encoder-decoder, but by fundamentally changing the trained parameters utilised in the model itself for generation and rendering. Many more potential hacks are available including more profound manipulation of text-to-audio models.

References

- [1] Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., et al., “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 12, 2016.

- [2] Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A., “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, 36, pp. 47704–47720, 2023.
- [3] Bergstrom, I. and Lotto, R. B., “Code Bending: A new creative coding practice,” *Leonardo*, 48(1), pp. 25–31, 2015.
- [4] Ghazala, R., *Circuit-Bending: Build your own alien instruments*, John Wiley & Sons, 2005.
- [5] Blackwell, A. F., Cocker, E., Cox, G., McLean, A., and Magnusson, T., *Live coding: a user’s manual*, MIT Press, 2022.
- [6] Bovermann, T. and Griffiths, D., “Computation as material in live coding,” *Computer music journal*, 38(1), pp. 40–53, 2014.
- [7] Broad, T., Leymarie, F. F., and Grierson, M., “Network bending: Expressive manipulation of generative models in multiple domains,” *Entropy*, 24(1), p. 28, 2022.
- [8] Yee-King, M., “Latent spaces: A creative approach,” in *The Language of Creative AI: Practices, Aesthetics and Structures*, pp. 137–154, Springer, 2022.
- [9] Collins, N., Ruzicka, V., and Grierson, M., “Remixing AIs: mind swaps, hybridity, and splicing musical models,” in *Proc. The Joint Conference on AI Music Creativity*, 2020.
- [10] Dzwonczyk, L., Cella, C. E., and Ban, D., “Network Bending of Diffusion Models for Audio-Visual Generation,” 2024.
- [11] Cascone, K., “The Aesthetics of Failure: ‘Post-Digital’ Tendencies in Contemporary Computer Music,” *Computer Music Journal*, 24(4), pp. 12–18, 2000.
- [12] Hegarty, P., *Noise/Music: A History*, Continuum, New York, 2007.
- [13] Evans, Z., Parker, J. D., Carr, C., Zukowski, Z., Taylor, J., and Pons, J., “Stable audio open,” *arXiv preprint arXiv:2407.14358*, 2024.
- [14] Evans, Z., Parker, J. D., Carr, C., Zuckowski, Z., Taylor, J., and Pons, J., “Long-form music generation with latent diffusion,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2024.
- [15] Cosgrove, S., “Techno! The New Dance Sound of Detroit (liner notes),” 1988.
- [16] Collins, N., Schedel, M., and Wilson, S., *Electronic music*, Cambridge University Press, 2013.
- [17] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O., “librosa: Audio and music signal analysis in python,” in *SciPy*, pp. 18–24, 2015.
- [18] Collins, N., “SCMIR: A SuperCollider music information retrieval library,” in *Proceedings of the International Computer Music Conference (ICMC)*, 2011.
- [19] Smalley, D., “Spectromorphology: explaining sound-shapes,” *Organised sound*, 2(2), pp. 107–126, 1997.
- [20] Ronchini, F., Comanducci, L., Perego, G., and Antonacci, F., “PAGURI: a user experience study of creative interaction with text-to-music models,” 2024.
- [21] Karimi Mahabadi, R., Henderson, J., and Ruder, S., “Compacter: Efficient low-rank hypercomplex adapter layers,” *Advances in Neural Information Processing Systems*, 34, pp. 1022–1035, 2021.
- [22] Plitsis, M., Kouzelis, T., Paraskevopoulos, G., Katsouros, V., and Panagakis, Y., “Investigating personalization methods in text to music generation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1081–1085, IEEE, 2024.
- [23] Ding, Y. and Lerch, A., “Parameter-Efficient Transfer Learning for Music Foundation Models,” *arXiv preprint arXiv:2411.19371*, 2024.
- [24] Li, E., Liu, J., and Mehra, G., “Efficient Fine-Tuning of CLAP-Guided Latent Stable Diffusion for High Fidelity Music Generation,” Technical report, Harvard, 2024.