

Latent Novelification: Expanding Generative Expressivity Through Objective-Defined Transformation

Joseph Meyer
Creative Computing Institute,
University of the Arts London
London, United Kingdom
j.meyer@arts.ac.uk

Mick Grierson
Creative Computing Institute,
University of the Arts London
London, United Kingdom
m.grierson@arts.ac.uk

Sarah Fdili Alaoui
Creative Computing Institute,
University of the Arts London
London, United Kingdom
s.fdilialaoui@arts.ac.uk

Nick Bryan-Kinns
Creative Computing Institute,
University of the Arts London
London, United Kingdom
n.bryankinns@arts.ac.uk

Rebecca Fiebrink
Creative Computing Institute,
University of the Arts London
London, United Kingdom
r.fiebrink@arts.ac.uk

Abstract

Musicians are using neural audio models to synthesize new sounds, but the sound space accessible through such models is often constrained to essentially interpolated approximations of existing training data, limiting the true sonic novelty musicians can elicit from generative models. To address this problem, we present Latent Novelification (LN), a machine learning method which learns to transform vectors in any generative model’s latent space into new, contrasting vectors according to an objective function which we define. The LN objective enforces that these contrasting vectors explicitly diverge from existing training data while remaining semantically meaningful to the generative model, facilitating generative novelty beyond simple interpolation. LN supports divergent creativity by helping musicians generate new and novel sounds, integrating into any neural audio model interface to controllably and interactively expand accessible sound space. This work contributes a new algorithmic method—implemented in open-source software—which musicians working with neural audio synthesis can use to synthesize new sounds to incorporate into their art.

CCS Concepts

• **Computing methodologies** → **Machine learning**; • **Human-centered computing** → **Human computer interaction (HCI)**; • **Applied computing** → *Sound and music computing*.

Keywords

Generative AI, Neural Audio Synthesis, Active Divergence, Network Bending, Latent Space, RAVE

ACM Reference Format:

Joseph Meyer, Mick Grierson, Sarah Fdili Alaoui, Nick Bryan-Kinns, and Rebecca Fiebrink. 2026. Latent Novelification: Expanding Generative Expressivity Through Objective-Defined Transformation. In *Creativity and Cognition (C&C '26)*, July 13–16, 2026, London, United Kingdom. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3803784.3816873>

C&C '26, London, United Kingdom

© 2026 Copyright held by the owner/author(s).

This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Creativity and Cognition (C&C '26)*, July 13–16, 2026, London, United Kingdom, <https://doi.org/10.1145/3803784.3816873>.

1 Introduction

Musicians want to synthesize novel sounds to incorporate into their work [17] [16] [21]. Neural audio models can facilitate this [28] [12], but interfaces for using them often map to limited subsets of their generative range closely aligned with or interpolating existing sounds [25] [34], limiting output novelty and divergent creative applications [4] [32]. Latent Novelification (LN) addresses this problem by applying active divergence principles [2] to find diverse sounds outside of a training distribution. Given a latent control space, LN learns a contrasting one which can be interactively interpolated with the original to expand accessible generative range, supporting divergent creative applications. We contribute a method for defining complex transformations of latent space, and provide an effective architecture and training procedure, open-source software implementation [23] integrated with RAVE [5], and Max interface using LN for movement sonification tuning.

2 Background

2.1 Generating Novel Neural Network Outputs

Prior work has explored elicitation of novel outputs from neural networks. Network bending [3] [19] inserts mathematical operations into a generative neural network so that it diverges from its training distribution—a practice termed “active divergence” [2]. LN is a more customizable form of network bending which isolates and manipulates objective-defined latent subspaces to generate novel content.

Creative Adversarial Networks (CANs) [9] optimize a generator network to produce output which a discriminator network classifies as “art” (as opposed to “not art”) but cannot confidently categorize into genre labels. LN similarly balances divergence and interpretability, but transforms embeddings instead of generating content directly. This provides different affordances, making it more useful for integrating into other generative models.

Novelty search [20] [18] optimizes a sparseness objective to find novel content. It has been used to generate novel embeddings, including for RAVE [13]. LN adapts these principles to transform existing control spaces, expanding the expressive range of any latent-based generative interface.

2.2 Neural Audio Models and Interfaces

Neural audio models generate raw waveforms or spectrograms. Many are autoencoder-based [11], mapping a latent control space to sound. RAVE [5] is an open-source implementation popular for digital instrument design [26] [33] [6].

Musicians are using neural audio models to generate new sounds for their art. Holly Herndon used ML to synthesize glitchy voices on her album “Proto” [15], and her gallery exhibition “The Call” [14]. Canblaster and Björk have also used neural synthesis in their work [7] [8].

Researchers have developed a variety of strategies for mapping human control to a neural audio model’s latent control space, enabling interactive manipulation of generated audio. Zheng et al. embedded existing sounds into interactively navigable terrain, using supervised learning to extrapolate sonic interpolations [34]. Vigliensoni and Fiebrink [30] used interactive ML to map interaction space to generative latent space. Meyer et al. [24] mapped embodied movement to RAVE’s latent space with unsupervised learning, via multimodal autoencoder. LN expands the expressive range of such systems, augmenting their control spaces to access novel sounds.

Researchers have also experimented with network bending neural audio models. Yee-King & McCallum [31] applied network bending to DDSF [10], proposing and analyzing a set of audio-specific bending operations. Manz and Grierson [19] applied traditional network bending to RAVE, enabling interactive audio manipulation via virtual sliders. LN extends this approach, using machine learning to define more complex custom bending functions mapping to contrasting latent subspaces, as described below.

3 LN Method

3.1 High-Level Explanation

Latent Novelification (LN) entails training a “novelifier” neural network to compute a “novelification” function N that maps any embedding vector in the generative model’s latent control space to a different, contrasting vector in the latent control space (Figure 1). Audio can then be generated by applying the generative model’s decoder to any novelified embedding vector. Further, because novelified and original embeddings are both vectors in the latent control space of the original generative model, they can be interactively interpolated between along any combination of dimensions, as shown in Figure 2.

The novelifier neural network is trained through unsupervised learning to maximize its output’s mathematical novelty relative to a “novelifier training set” of reference embeddings. The following sections detail LN objectives and training procedure.

3.2 Objective Functions

In this section, we define LN training objectives for divergence and stability.

3.2.1 Divergence Objectives. A novelified sound should be different from its original non-novelified sound, from other non-novelified sounds, from other novelified sounds, and from silence. These criteria are reflected, respectively, in the four objectives below, for a novelification function N , audio encoder E , and non-novelified

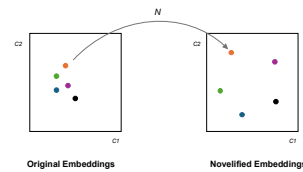


Figure 1: LN learns a mapping function N from an existing latent control space to a contrasting one. N transforms original embeddings (shown here as points in 2D space with latent dimensions $C1$ and $C2$), moving them around within a generative model’s latent space.

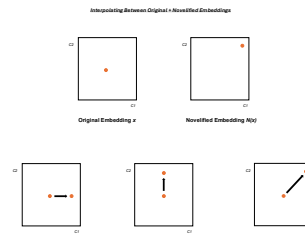


Figure 2: A user can apply the mapping function N directly to any embedding (top). Or, the user can interpolate between the original and the novelified embedding along each latent dimension, or along a combination of latent dimensions (bottom).

embeddings x and y randomly sampled from novelifier training data:

- **Transformation Depth** := $R(N(x), x)$;
Magnitude of LN transformation, i.e. the difference before and after novelification
- **Novelty** := $R(N(x), y)$;
Difference between novelified embeddings and other un-novelified embeddings
- **Diversity** := $R(N(x), N(y))$;
Difference between pairs of novelified embeddings
- **Audibility** := $R(N(x), E(0))$;
Difference between novelified embeddings and encoded silence

Using empirical experimentation guided by our intuition and aesthetic preferences¹, we designed a custom reward function R to use in these objective functions, which increases quickly when similar embeddings start to diverge, and more slowly as embeddings become further apart (illustrated in red in Figure 3). This reward is smallest for identical embeddings, and progressively larger as embeddings become more distant.

In particular, for audio embeddings a and b (either of which may be novelified, or not), we define this reward function²:

$$R(a, b) := R'(a, b) + R'(\text{abs}(a), \text{abs}(b))$$

¹starting with simple distance metrics and building out complexity incrementally based on subjective evaluation of effects on audio output

²MAE is mean absolute error, and k is a configurable parameter indicating instability tolerance (higher k will yield more divergence relative to stability). Based on empirical experimentation, the default instability tolerance is set to $k = 1.2$.

where

$$R'(a, b) := k \cdot \log(\text{MAE}(a, b) + 1) - \frac{1}{\sqrt{\text{MAE}(a, b) + 1}}$$

3.2.2 Stability Objective. To avoid uninteresting output (e.g. noise), novelified embeddings should also capture semantically meaningful audio features, which we enforce by proxy as reproducibility by the generative model:

- **Stability** := $-\text{MSE}(N(x), E(D(N(x))))$;
How unchanged a novelified embedding is when decoded to audio and re-encoded back to latent space

for novelifier N , un-novelified embedding x , decoder model D , and encoder model E .

As in the divergence objective computation, we compute this function for random samples of x from the novelifier training set during the novelifier training optimization process.

Illustrated in Figure 3 in blue, this stability objective function is designed to yield low stability incentive (i.e. slope) when $N(x)$ changes little across decoding and re-encoding, and high stability incentive when decoding and re-encoding $N(x)$ yields a vastly different embedding $E(D(N(x)))$. Stability reward (i.e., $f(x)$) is highest when novelified embeddings do not change at all when decoded and re-encoded, and progressively lower the more decoding and re-encoding embeddings changes them.

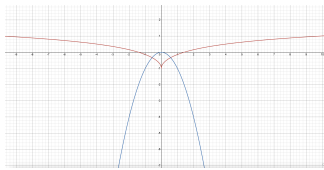


Figure 3: Graph of the divergence reward function R' (red) and stability objective (blue) computed as $f(x)$ where $x := z - z'$ for embeddings z, z' . Both functions are symmetric across the y-axis. Traversing away from $x = 0$, the divergence reward rises sharply then levels off, while the stability reward declines gradually then steepens. Thus the divergence incentive overpowers the stability incentive as $x \rightarrow 0$, and as $x \rightarrow \pm\infty$ the stability incentive dominates.

3.3 Novelifier Architecture and Training Procedure

Our novelifier architecture, illustrated in Figure 4, consists of three neural network modules: a divergence maximization module, a stability maximization module, and a supermodule. The input to the divergence maximization module and stability maximization module is an un-novelified embedding vector, x . Each of these modules outputs an intermediary vector of the same length as the original embedding, i.e., $\text{len}(x)$. The supermodule takes an input of size $3 \times \text{len}(x)$; its input is a concatenation of the original embedding with the intermediary embeddings produced by the first two modules. The supermodule outputs the novelified embedding vector whose length is the same as the original, i.e. $\text{len}(x)$.

Author pre-print

During training, each of these modules' parameters are adjusted via three parallel optimizers, each with a different objective function. A divergence optimizer trains the divergence module to yield embeddings that, when processed by the supermodule, maximize the sum of the divergence objectives. A stability optimizer trains the stability module to yield embeddings that, when processed by the supermodule, maximize the stability objective. The super-optimizer trains the supermodule and the submodules to optimize both divergence and stability. During training, gradient updates alternate between these three optimizers.

The architecture of the individual neural networks and the choice of optimizer can be varied. For our base implementation, each of the three neural networks consists of two linear layers separated by ReLU activation trained using Adam optimization and an initial learning rate of 0.001. We partition data into train and validation sets, and halt training once the validation set objective stops improving, saving the model with the best validation set metrics.

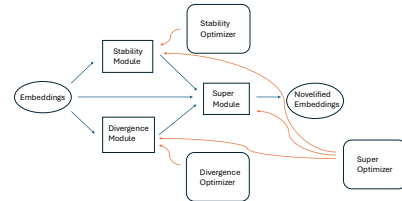


Figure 4: Novelifier architecture and training framework. Input embeddings flow to parallel divergence and stability modules, which both feed into a supermodule which outputs novelified embeddings. A divergence optimizer optimizes the divergence module, a stability optimizer optimizes the stability module, and a super-optimizer optimizes all modules.

4 LN Movement Sonification Interface

We integrated LN into a movement sonification system [24], and built a Max interface, shown in Figure 5, for users to interactively manipulate the latent control space their movements map to. Human pose keypoints (i.e. skeletal coordinates of a human body) are extracted from live laptop camera feed, and mapped to RAVE's latent audio space using autoencoders trained on custom poses and audio files [24]. LN is applied after mapping pose keypoints to latent audio space, before decoding to audio.

We created three interface features:

- **“Novelification Ratio”** slider controls the ratio of interpolation between novelified and un-novelified embeddings, from 0 (all un-novelified) to 1 (all novelified)
- **“Latent Gain”** slider controls magnitude of final latent embedding (continuous scaling from $0x$ to $2x$)

- **“Toggle Novelification Mode”**³ toggles between three modes: fully novelified, only first dimension novelified, programmable subset of dimensions novelified according to specified ratios

Three live multislider displays visualize weighted un-novelified and novelified embeddings, and final mixed embedding.

“Start/Stop” button upon odd press enables RAVE decoder inference, activates audio playback and recording; upon even press disables RAVE decoder inference, turns off audio playback, and saves recording.

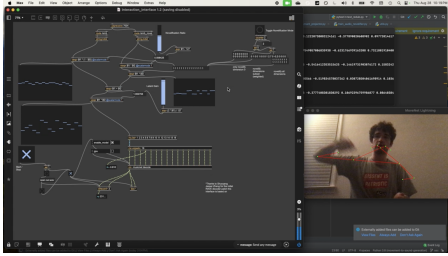


Figure 5: User Interface for Movement Sonification with Latent Novelification. A programmable button toggles novelification mode (i.e. the relative proportions of novelification for each latent dimension), and two additional sliders control novelification ratio and latent gain. Original embeddings are computed by mapping from human movement, i.e. pose key-points (shown as green dots superimposed over the person moving).

5 Discussion

Latent Novelification may help improve the expressive range and interactivity of neural audio systems, supporting divergent creativity by enabling musicians to generate unique, interesting sounds. In a demo video⁴, we show how LN interactively changes audio timbre⁵ [22]. We have also used LN to create a musical composition⁶ [1], finding that the additional generative control it offered helped us synthesize sounds we preferred, validating its artistic applicability.

LN may be extended by matching objectives to human perceptual range (e.g. removing non-human-audible frequencies from generated audio during training), interactive fine-tuning (e.g. letting users reject undesirable transformations by adding their output to novelifier training set), or network bending the novelifier itself. LN also applies to other mediums beyond audio, including visual, text, and choreographic modes. Its objective functions, neural architecture, and training procedure can be customized and made

³This interface uses a toggle button for novelification control to encourage users to manipulate sound through their movement rather than tweaking sliders. The design frames novelification control as a mechanism to adjust the sound space the movement can interactively generate from, similar to programmable mode presets on a keyboard synthesizer.

⁴using LN trained on a few minutes of Philharmonia violin data [27] to manipulate Bach fugues autoencoded by a RAVE model pre-trained on MusicNet data [29]

⁵e.g. altering frequency content and making sounds “smoother” and “bassier”

⁶In collaboration with musician Alessandro Bonomi (https://open.spotify.com/artist/127aCDkq9H9veeKQQiEcb!si=CnEA_4yIRuiOWNjg3bN6hw)

arbitrarily more complex. LN is a powerful tool with the potential to make generative models more expressive, interactive, and artistically useful.

6 Conclusion

We presented Latent Novelification, a new method for transforming latent space and improving generative expressivity. We conveyed the motivation and theory behind LN, contextualized it within existing literature, and explained how it works. We offered an effective, open-source LN implementation, integrated into RAVE, along with a Max interface for tuning movement sonification mappings. We demonstrated how LN may usefully augment generative musical interfaces to help artists synthesize novel sounds.

Acknowledgments

Thanks to Shuoyang Zheng; our Max patches are based on one he created.

References

- [1] Alessandro Bonomi and Joseph Meyer. 2026. Aquatic rift. (2026). <https://ualres.earchonline.arts.ac.uk/id/eprint/26709/>.
- [2] Terence Broad, Sebastian Berns, Simon Colton, and Mick Grierson. 2021. Active divergence with generative deep learning – a survey and taxonomy. (July 12, 2021). arXiv: 2107.05599[cs]. doi:10.48550/arXiv.2107.05599.
- [3] Terence Broad, Frederic Fol Leymarie, and Mick Grierson. 2021. Network bending: expressive manipulation of deep generative models. (Mar. 12, 2021). arXiv: 2005.12420[cs]. doi:10.48550/arXiv.2005.12420.
- [4] Nick Bryan-Kinns et al. 2025. Leveraging small datasets for ethical and responsible ai music making. In *Proceedings of the 20th International Audio Mostly Conference (AM '25)*. Association for Computing Machinery, 70–81. ISBN: 9798400720659. doi:10.1145/3771594.3771601.
- [5] Antoine Caillon and Philippe Esling. 2021. RAVE: a variational autoencoder for fast and high-quality neural audio synthesis. (Dec. 15, 2021). arXiv: 2111.05011[cs]. doi:10.48550/arXiv.2111.05011.
- [6] Isaac Clarke, Francesco Ardan Dal Ri, 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 Article 32*. Doga Cavdir and Florent Berthaut, (Eds.) Canberra, Australia, 224–230. doi:10.5281/zenodo.15735662.
- [7] Nils Demerlé, Philippe Esling, Guillaume Doras, and David Genova. 2024. After: audio features transfer and exploration in real-time. (2024). <https://github.com/acids-ircam/AFTER>.
- [8] Nils Demerlé, Philippe Esling, Guillaume Doras, and David Genova. 2024. Combining audio control and style transfer using latent diffusion. In *International Society for Music Information Retrieval, ISMIR 2024*. <https://arxiv.org/abs/2408.00196>.
- [9] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. CAN: creative adversarial networks, generating “art” by learning about styles and deviating from style norms. (June 21, 2017). arXiv: 1706.07068[cs]. doi:10.48550/arXiv.1706.07068.
- [10] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. 2020. Ddsp: differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*.
- [11] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International conference on machine learning*. PMLR, 1068–1077. <https://proceedings.mlr.press/v70/engel17a/engel17a.pdf>.
- [12] Corey Ford, Ashley Noel-Hirst, Sara Cardinale, Jackson Loth, Pedro Sarmiento, Elizabeth Wilson, Lewis Wolstanholme, Kyle Worrall, and Nick Bryan-Kinns. 2024. Reflection across ai-based music composition. In *Proceedings of the 16th Conference on Creativity & Cognition*, 398–412. <https://dl.acm.org/doi/pdf/10.1145/3635636.3656185>.
- [13] Jinyue Guo, Anna-Maria Christodoulou, Balint Laczko, and Kyrre Glette. 2024. Lvns-rave: diversified audio generation with rave and latent vector novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 667–670.
- [14] Holly Herndon and Mat Dryhurst. 2024. Holly herndon & mat dryhurst: the call. (2024). <https://www.serpentinegalleries.org/whats-on/holly-herndon-mat-dryhurst-the-call/>.
- [15] Holly Rebecca Herndon. 2019. *Proto*. Stanford University.

- [16] Barry Hill. 2005. Breaking down the breakdown: the use of timbres in contemporary dance music sub-genres. In *Proceedings of the Australasian Computer Music Conference (ACMC 2005)*. <https://computermusic.org.au/proceedings/ACMC2005-proceedings/acmc05-075-084.pdf>.
- [17] Björn Þór Jónsson, Amy K. Hoover, and Sebastian Risi. 2015. Interactively evolving compositional sound synthesis networks. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO '15)*. Association for Computing Machinery, Madrid, Spain, 321–328. ISBN: 9781450334723. doi:10.1145/2739480.2754796.
- [18] Joel Lehman, Kenneth O Stanley, et al. 2008. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*. Vol. 65, 329–336.
- [19] Daniel Manz and Mick Grierson. 2025. Brave: designing an embedded network-bending instrument, manifesting output diversity in neural audio systems. https://ualresearchonline.arts.ac.uk/id/eprint/24221/1/manz_grierson_camer_a_ready.pdf.
- [20] Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83, 12, 2481–2497.
- [21] Adnan Marquez-Borbon and Paul Stapleton. 2015. Fourteen years of nime: the value and meaning of community in interactive music research. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 307–312.
- [22] Joseph Meyer. 2026. Ln movement sonification interface demonstration video. (2026). <https://youtu.be/cr9FXQgDNzw>.
- [23] Joseph Meyer. 2026. Tambereen. (2026). <https://github.com/joemeyer1/tambereen>.
- [24] Joseph Meyer, Nick Bryan-Kinns, Sarah Fdili Alaoui, Mick Grierson, and Rebecca Fiebrink. 2025. Interactive movement-to-audio with pre-trained neural networks. In *Proceedings of the 2025 Conference on Creativity and Cognition*. C&C '25: Creativity and Cognition. ACM, Virtual United Kingdom, (June 23, 2025), 491–493. ISBN: 979-8-4007-1289-0. doi:10.1145/3698061.3734415.
- [25] Sarah Nabi, Nils Demerlé, Geoffroy Peeters, Frederic Bevilacqua, and Philippe Esling. 2025. Adding temporal musical controls on top of pretrained generative models. In *Ismir 2025 Hybrid Conference*. <https://doi.org/10.5281/zenodo.17811482>.
- [26] Sarah Nabi, Philippe Esling, Geoffroy Peeters, and Frédéric Bevilacqua. 2024. Embodied exploration of deep latent spaces in interactive dance-music performance. In *Proceedings of the 9th International Conference on Movement and Computing*. ACM, Utrecht Netherlands, (May 2024), 1–9. ISBN: 9798400709944. doi:10.1145/3658852.3659072.
- [27] Philharmonia. 2017. Philharmonia sound samples. Accessed 2025. (2017). <https://philharmonia.co.uk/resources/sound-samples/>.
- [28] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design Principles for Tools to Support Creative Thinking. <https://doi.org/10.1184/R1/6621917.v1>.
- [29] John Thickstun, Zaid Harchaoui, and Sham Kakade. 2017. Learning features of music from scratch. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=rkFBjv9gg>.
- [30] Gabriel Vigliensoni and Rebecca Fiebrink. 2023. Steering latent audio models through interactive machine learning. https://ualresearchonline.arts.ac.uk/id/eprint/20199/1/VigliensoniFiebrink_ICCC2023.pdf.
- [31] Matthew Yee-King and Louis McCallum. 2021. Studio report: sound synthesis with ddsp and network bending techniques. In *Proceedings of 2nd Conference on AI Music Creativity*. AIMC. <https://ualresearchonline.arts.ac.uk/id/eprint/20629/6/820d0b5d-bbaf-4d24-bce5-c9e2cbe8fad6.pdf>.
- [32] Weitao Zhang, Zsuzsika Sjoerds, and Bernhard Hommel. 2020. Metacognition of human creativity: the neurocognitive mechanisms of convergent and divergent thinking. *NeuroImage*, 210, 116572. doi:<https://doi.org/10.1016/j.neuroimage.2020.116572>.
- [33] Shuoyang Zheng, Anna Xambó Sedó, and Nick Bryan-Kinns. 2024. A Mapping Strategy for Interacting with Latent Audio Synthesis Using Artistic Materials. arXiv:2407.04379. (July 2024). doi:10.48550/arXiv.2407.04379.
- [34] Shuoyang Zheng, Bleiz M Del Sette, Charalampos Saitis, Anna Xambó, and Nick Bryan-Kinns. 2024. Building sketch-to-sound mapping with unsupervised feature extraction and interactive machine learning. In *Proceedings of the International Conference on New Interfaces for Musical Expression* Article 86. S M Astrid Bin and Courtney N. Reed, (Eds.) Utrecht, Netherlands, (Sept. 2024), 591–597. doi:10.5281/zenodo.13904959.