# Building Sketch-to-Sound Mapping with Unsupervised Feature Extraction and Interactive Machine Learning

Shuoyang Zheng
Centre for Digital Music
Queen Mary University of London
shuoyang.zheng@qmul.ac.uk

Bleiz M. Del Sette
Centre for Digital Music
Queen Mary University of London
b.delsette@qmul.ac.uk

Charalampos Saitis
Centre for Digital Music
Queen Mary University of London
c.saitis@qmul.ac.uk

Anna Xambó
Centre for Digital Music
Queen Mary University of London
a.xambosedo@qmul.ac.uk

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

## ABSTRACT

In this paper, we explore the interactive construction and exploration of mappings between visual sketches and musical controls. Interactive Machine Learning (IML) allows creators to construct mappings with personalised training examples. However, when it comes to high-dimensional data such as sketches, dimensionality reduction techniques are required to extract features for the IML model. We propose using unsupervised machine learning to encode sketches into lower-dimensional latent representations, which are then used as the source for the IML model to construct sketch-to-sound mappings. We build a proof-of-concept prototype and demonstrate it using two compositions. We reflect on the composing processes to discuss the controllability and explorability in mappings built by this approach and how they contribute to the musical expression.

## Author Keywords

Cross-modal mapping, unsupervised learning, variational autoencoder, sound synthesis control

## CCS Concepts

•**Applied computing** → **Sound and music computing;** •**Human-centered computing** → *Graphics input devices;* •**Computing methodologies** → Neural networks;

## 1. INTRODUCTION

Sketching is an intuitive and natural form of communication, and there has been extensive research on using sketches as sound control interfaces within the NIME community. These works have been used for a variety of applications, including cross-modal control of sound synthesis [17], composition [1, 5], annotation [28], melody generation [13, 22], and graphic sonification [3, 23, 31]. Cross-modal studies have shown that meaningful perceptual associations exist between shapes and sounds [15], but these associations in most sketch-to-sound applications are pre-determined by instrument makers, and fixed for all musicians who use them [25]. However, a musician may seek to personalise these shape-sound mappings for specific creative goals [12, 20]. Therefore, in this paper we aim to explore the potential of using interactive machine learning to help musicians build personalised sketch-to-sound mappings.

Interactive Machine Learning (IML) [7] is commonly used to create small-scale tailored mapping models. In the domain of music, it is often used to build mapping between sensor inputs and sound controls [8]. However, when it comes to complex high-dimensional inputs, such as sketches, feature extraction techniques are required to encode these inputs into lower-dimensional representations to be used as the source of the IML model.

Previous works [2, 30] tackling image information retrieval have shown unsupervised feature learning's good capability in extracting representative features from a corpus of unlabeled data. However, with a few exceptions [21, 26], there have not been many works on using this technique to create expressive musical mappings. Therefore, we focus on exploring unsupervised feature learning to build mappings between visual sketches and sound controls. The work presented in this paper is driven by two research questions:

1. How can we leverage unsupervised feature learning for IML-based sketch-to-sound mapping?
2. What are the unique interaction experience of sketch-to-sound mapping built with this approach?

We present the process of integrating unsupervised feature extraction for IML to build our sketch-to-sound controller, and explain our system design considerations during the development process. Then, we demonstrate the controller with two performance sets composed by the first author. We reflect on their experience during the composition process to discuss findings about our controller's controllability and explorability.

## 2. RELATED WORK

Various strategies have been proposed to tackle feature extraction for sketch-to-sound mappings. Low-level features



*NIME'24,* 4–6 September, Utrecht, The Netherlands.

such as the position [23, 31] and trajectory of sketches [13, 22] are useful sources that can be mapped to sound controls. Further, supervised machine learning allows the feature extraction model to recognise a set of shapes [5] or higher-level characteristics in a sketch such as *noisy* and *calm* [14]. Works using supervised machine learning, such as *SketchSynth* [17], have been shown to yield sketch-sound associations that are close to those provided by humans. However, a major challenge of these approaches is to define meaningful features that are useful for constructing mapping between sketches and sounds [15].

## 2.1 Unsupervised Feature Learning

In contrast, unsupervised feature learning allows the feature extraction model to learn representative features from a large corpus of unlabelled data [2]. The trained model encodes high-dimensional inputs into lower-dimensional latent representations. These latent representations can be seen as a compressed format of the original input. In the domain of music, unsupervised feature learning is often used to create open-ended mappings. For example, Roma et al. [26] apply unsupervised dimensionality reduction on sound collections to create interactive sound spaces. Further, Murray-Browne and Tigas [21] use unsupervised feature extraction as a mapping between sensor inputs and lower-dimensional latent representations, which are then used to control a synthesiser's parameters. Open-ended mappings offer a new perspective that allows performers to embrace unplanned outputs in a musical instrument [4]. Following this direction, our work explores how this level of openness provided by unsupervised methods can be applied between sketch-to-sound mappings.

## 2.2 Interactive Machine Learning

Interactive Machine Learning (IML) [7] allows creators to construct mappings between human control space and sound synthesis parameters using a few personalised training examples [6]. It focuses on the mapping model's incremental construction process, in which creators iteratively record paired inputs and desired output, enabling them to build relationships between the two spaces [29] and experiment new approaches to sound synthesis [20]. Tools encapsulating IML, such as *Wekinator* [9] and *Learner.js* [19] have been widely used to facilitate the model training process. Our work uses IML to connect extracted sketch features and synthesis parameters.

# 3. DEVELOPMENT

This section aims to address the first research question by explaining the design and development process of our sketch-to-sound music controller. A high-level system diagram is shown in Figure 1. We use an unsupervised feature extraction model to encode sketches into latent representations, then use it as the source for the interactive machine learning model to build the sketch-to-sound mapping. The following sections describe these two parts in detail.

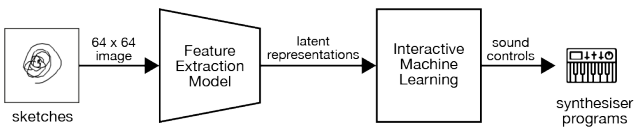


**Figure 1: A high-level sketch-to-sound system diagram**

## 3.1 Unsupervised Feature Learning with VAE

A common approach for unsupervised feature learning is using Variational Autoencoders (VAE) [30], an unsupervised learning model that comprises an encoder and a decoder. After being trained on a corpus of unlabelled data, the encoder maps new incoming data to a point in a Gaussian distribution. This data point is a low-dimensional latent representation, which is then used by the decoder to reconstruct the original input. VAE relies on deep neural networks for visual data. However, integrating such models in a real-time performing system can be complex and problematic.

### *3.1.1 Optimising the VAE Model*

Firstly, the system's functionality would be largely determined by the accuracy and diversity of the deep learning model. Specifically, it needs to generate a latent space that can accurately reconstruct the input sketches, and it also needs to respond to a diverse input set without diverging and overfitting. With regard to this issue, we used the Deep Feature Consistent Variational Autoencoder (DFC-VAE) [11], which is a variation of VAE that replaces the pixel-by-pixel loss by perceptual loss computed by a pre-trained VGG19 network [27]. This ensures the sketch feature extraction model provides a latent space with better perceptual quality.

Secondly, a deep learning model usually requires heavier computational power to ensure real-time functionality [10]. We attempted to limit the number of parameters in our model to keep it as lightweight as possible. By scaling the input resolution to $64 \times 64$, we reduced the model's size while ensuring the image can capture enough details in a sketch. We reduced the number of hidden layers and the latent representation's dimension to 5 and 32 respectively, which are the lowest numbers we can get while keeping the model with comparable quality.

After training the model for 10k steps, we used only the encoder to compress sketches to latent representations. The encoder was calibrated by setting the latent representation of empty frames to zeros, and using the differences between new inputs and the empty frame as calibrated outputs. Therefore, the latent representation of an empty frame is initialised to zeros. The trained model is deployed on a separate device with an RTX 4060 laptop GPU and can run at a maximum of 20 frames per second.

### *3.1.2 Data Augmentation*

In addition, we attempted to increase the data diversity of our feature extraction model by implementing a data augmentation process with random rotations and shifts. The *Sketching Sounds* dataset created by Löbbers et al. [16] is used to train the VAE. It contains sketch data in image format with their descriptors. However, our training only uses sketch data because it is an unsupervised approach that does not require labelled descriptors.

## 3.2 Steering Latent Representations with IML

We connected the feature extraction model trained in the previous section to an Interactive Machine Learning (IML) model. We used the Wekinator, a commonly used tool for IML that allows creators to record training examples for the training [9]. Our Wekinator model is a composition of 8 regressive neural network models, taking the latent representations from the feature extraction model as their input. In order to map the Wekinator model's output to sound con-

**Figure 2: A screenshot of the Max4Live receiver device (the one on the leftmost) and the sender device (the other eight).**

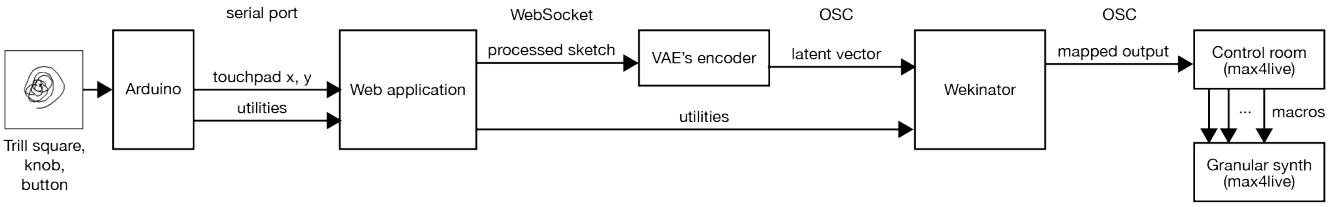


**Figure 3: A detailed flowchart of system components in the sketch-to-sound pipeline.**

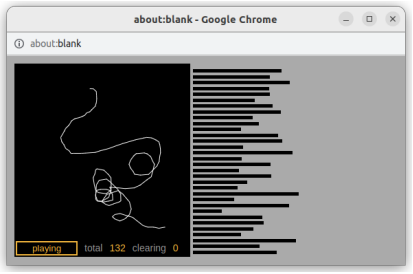


**Figure 4: The web interface that visualise the sketch and its latent representations**

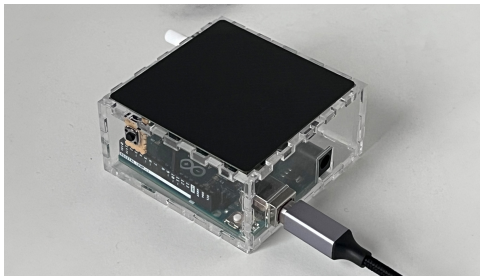

**Figure 5: The physical controller with a touchpad, a button, and a knob.**

trol parameters, they are hard clamped into the range of [0, 1], and then sent as OSC messages. We built a Max4Live patch shown in Figure 2 (left) with 8 sliders to connect the Wekinator model with Ableton Live. They are used as macro controls, which are a set of parameters that can be mapped to other sound synthesis programs using the sender devices shown in Figure 2 (right).

## 3.3 Implementation

Our detailed implementation is illustrated in Figure 3. We built a web application shown in Figure 4 to visualise the sketch and its latent representations. For a compact and tangible experience, we built a physical controller shown in Figure 5 with a knob, a button, and a touchpad which is a Bela Trill Square[1] sensor. The knob controls how long a sketch will stay on the canvas. The button facilitates the recording of training examples for the Wekinator model. When pressed, the current sketch and macro controls are marked as training data and sent to the Wekinator.

The physical controller runs on an Arduino hardware, sending sensor data to the web application through serial port. The web application processes sensor data into an image format sketch. It is deployed from a Flack[2] backend program running in a Python environment, which is also where the DFCVAE encoder is running. The encoded latent representation are sent as OSC messages via the WebSocket protocol.

The source code for our Arduino program, training code for the DFCVAE model, Python backend scripts, web application, and the Max4Live devices can be accessed at our GitHub repository[3].

[1] `https://github.com/BelaPlatform/Trill-Arduino`
[2] `https://flask.palletsprojects.com/en/3.0.x/`

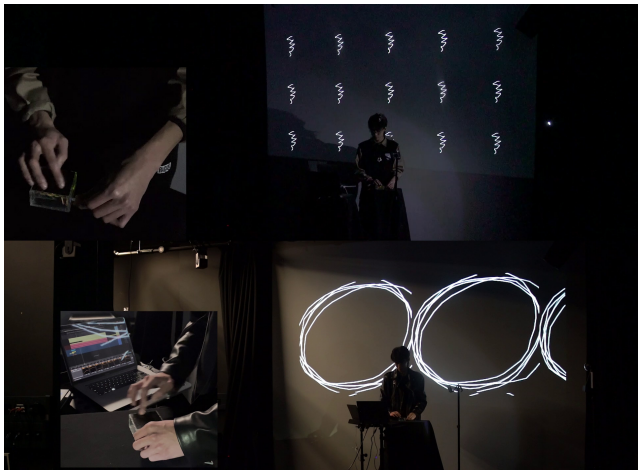

**Figure 6: Two performances using our sketch-to-sound controller. Video recordings can be viewed at `https://vimeo.com/907654328` and `https://vimeo.com/907654507`.**

## 4. COMPOSITIONS

We present two compositions built and performed by the first author using our sketch-to-sound controller. In order to address our second research question, we use these two

[3] `https://github.com/jasper-zheng/unsupervised-sketch-to-sound`

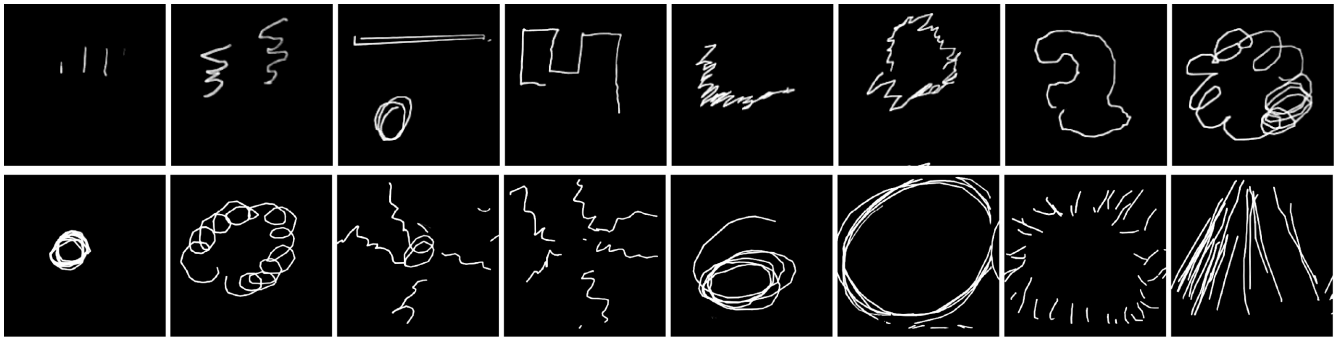

**Figure 7: Screen recording showcasing examples of sketches used during the performances.**

compositions as a first-person investigation of the underlying qualities and characteristics of the mapping. The first author kept a self-report journal during the composing, pre-performance and post-performance phases of the pieces to record their experience. Performances of these two compositions are shown in Figure 6 with links to the video recordings. The full written journal can be found in Appendix A.

In the first composition, the first author used ten granular synthesisers to create a 6-minute performance set. They used different sketch patterns to trigger different portions of samples loaded in these granular synthesisers. During the performance, the controller is played along with a programmed soft and slow chord progression, loosely triggering samples to create an ambient environmental soundscape. Automation was moderately used to transit between sections. A few examples of sketches used in this composition are shown in the first row of Figure 7.

In the second composition, all the eight macro controls are mapped into parameters in a Serum wavetable synthesiser[4]. A sustained drone sound was produced and manipulated by sketches. In this 3-minute performance, no automation was programmed to the synthesiser parameters, all variations in sound were controlled by the sketch-to-sound mapping. A few examples of sketches used in this composition are shown in the second row of Figure 7.

# 5. REFLECTION

This section reflects on the journal mentioned in Section 4 to discuss the experience of using our controller. We found two main characteristics that are specifically related to unsupervised feature extraction.

## 5.1 Movement Sensitivity

Firstly, we observed that latent-based mappings are sensitive to small movements in the input. As the latent representations can be seen as a compressed format of high-dimensional data [24], these representations are continually changing with the sketches. Therefore, our approach is more sensitive than other sketch-to-sound applications in a way that small movements in the sketches can lead to changes in the sound controls. Compared with SketchSynth [17], which allows performers to gradually and carefully manipulate a timbre, the performer using our controller hardly focused on refining details in their sketches to finetune the sound output, instead, they discovered repertoires and unique ways to perform with the system. For example, in the first composition, the performer used the movement of the sketches as a way to jittering the granular window and creating randomised variations in the performance. In the second composition, the performer used the sensitivity to create a constantly evolving soundscape. And they were no longer actively seeking movements that trigger sonic responses. Instead, they aim to restrict their movements to a repeating pattern and maintain the overall shapes for a more stable sound.

There are both positive and negative sides to the performer's experience. On the positive side, this sensitivity means more intense sonic feedback is provided to the performer. As described in the journal for the first composition, "[the random percussive sounds] made me feel that the sketches were actually triggering something". This suggests to us that unsupervised latent mapping can be a useful tool for creating embodied performing experiences that have strong sonic feedback to the body movements [18]. On the negative side, this sensitivity provides less precise controls and introduces a lot of unpredictability to the system. As described in the journal for the first composition, "... it was easier to trigger specific sounds when there were only three to four sets of sounds. But once it went more than that, it became impossible to precisely trigger a specific sound. In the end I have to add automation to ensure composability". Similarly, negative and confusing feelings are also shown in the second composition due to increased difficulties in steering the sound.

## 5.2 Open-Ended Exploration Space

Secondly, we found that this layer of unsupervised latent mapping distorts the exploration space of the IML model in interesting ways. In IML – as well as other mapping approaches that require pre-defined features – the exploration space is usually decided by the creator. Although new mappings can always be iteratively inserted into existing ones, they still require the creator to have a clear vision of desired sound-shape associations. By contrast, unsupervised mapping approaches allow creators to start with little sense of the mapping [21] and then interactively discover new ones. Therefore, integrating unsupervised models into mappings built with purely supervised approaches can open an exploration space for surprising mappings that are not planned by the creator. This openness can be beneficial in a musical process [4]. For example, in the first composition, the performer discovered during improvisation that drawing horizontal straight lines triggers a vocal sample slice, which was not previously planned when building the mapping. This ended up forming a new section in the composition.

Moreover, when recording Wekinator's training examples, all synthesiser parameters remain static and unchanged over time. But when the system starts running, these parameters start changing and evolving as the sketch is being drawn and fading out. And the static system becomes a

[4] `https://xferrecords.com/products/serum/`

dynamic system with movements. Therefore, as described in the journal for the second composition, "... it's hard to foresee what it would sound like until the system actually starts running". This creates a gap between the mapping's construction stage and the using stage, forcing the musician to explore this unplanned dynamic after the mapping is constructed.

## 6. CONCLUSIONS AND FUTURE WORK

This paper presented our exploration of interactive constructions of mappings between visual sketches and musical controls. We presented an implementation that shows how to leverage unsupervised feature learning for IML-based sketch-to-sound mapping. We found evidence that the movement sensitivity and an open-ended exploration space afforded by this approach can bring meaningful movement-based interactions and surprising results of mapping between sketches and sound to the performer. Our reflection is based on a first-person perspective. While it requires a more in-depth user study with musicians in future work to investigate how unsupervised feature learning and IML-based mapping can be combined in more generic contexts, our current demonstrations suggest that this is a promising approach that can serve as an alternative technology option for sketch-to-sound-mapping.

## 7. ACKNOWLEDGMENTS

Shuoyang Zheng and Bleiz M. Del Sette are research students at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1].

## 8. ETHICS STATEMENT

The compositions and the reflections in this work used a first-person investigation that did not involve any human participants other than the first author themselves. The physical controller is made with the consideration of sustainability.

## 9. REFERENCES

[1] B. Banar and S. Colton. Connecting Audio and Graphic Score Using Self-supervised Representation Learning - A Case Study with György Ligeti's Artikulation. In *International Conference on Innovative Computing and Cloud Computing*, 2022.

[2] Z. Cao, X. Li, Y. Feng, S. Chen, C. Xia, and L. Zhao. ContrastNet: Unsupervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral imagery classification. *Neurocomputing*, 460:71–83, 2021.

[3] N. d'Alessandro and T. Dutoit. HandSketch Bi-Manual Controller Investigation on Expressive Control Issues of an Augmented Tablet. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 78–81, New York City, NY, United States, June 2007. Pages: 78–81 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[4] T. Dannemann, N. Bryan-Kinns, and A. McPherson. Self-Sabotage Workshop: a starting point to unravel sabotaging of instruments as a design practice. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 70–78, May 2023. Place: Mexico City, Mexico.

[5] H. Diao, Y. Zhou, C. A. Harte, and N. Bryan-Kinns. Sketch-Based Musical Composition and Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 569–572, London, United Kingdom, June 2014. Pages: 569–572 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[6] J. J. Dudley and P. O. Kristensson. A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems*, 8(2):8:1–8:37, June 2018.

[7] J. A. Fails and D. R. Olsen. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45, Miami Florida USA, Jan. 2003. ACM.

[8] R. Fiebrink, P. R. Cook, and D. Trueman. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 147–156, New York, NY, USA, May 2011. Association for Computing Machinery.

[9] R. Fiebrink, D. Trueman, and P. R. Cook. A Meta-Instrument for Interactive, On-the-Fly Machine Learning. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 280–285, Pittsburgh, PA, United States, 2009. ISSN: 2220-4806.

[10] B. Hayes, C. Saitis, and G. Fazekas. Neural Waveshaping Synthesis, July 2021. arXiv:2107.05050 [cs, eess].

[11] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep Feature Consistent Variational Autoencoder, Oct. 2016. arXiv:1610.00291 [cs].

[12] S. Jordà. Instruments and Players: Some Thoughts on Digital Lutherie. *Journal of New Music Research*, 33(3):321–341, Sept. 2004. Publisher: Routledge _eprint: https://doi.org/10.1080/0929821042000317886.

[13] T. Kitahara, S. Giraldo, and R. Ramírez. JamSketch: A Drawing-based Real-time Evolutionary Improvisation Support System. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 505–506, Copenhagen, Denmark, June 2017. Pages: 505–506 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[14] S. Lobbers and G. Fazekas. SketchSynth: a browser-based sketching interface for sound control. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 637–641, May 2023. Place: Mexico City, Mexico.

[15] S. Löbbers, M. Barthet, and G. Fazekas. Sketching sounds: an exploratory study on sound-shape associations, July 2021. arXiv:2107.07360 [cs, eess].

[16] S. Löbbers and G. Fazekas. Sketching Sounds Dataset, June 2023. 10.5281/zenodo.7590916.

[17] S. Löbbers, L. Thorpe, and G. Fazekas. SketchSynth: Cross-Modal Control of Sound Synthesis. In C. Johnson, N. Rodríguez-Fernández, and S. M. Rebelo, editors, *Artificial Intelligence in Music, Sound, Art and Design*, volume 13988, pages 164–179. Springer Nature Switzerland, Cham, 2023. Series Title: Lecture Notes in Computer Science.

[18] M. Mainsbridge. Feeling movement in live electronic music: An embodied autoethnography. In *Proceedings of the 8th International Conference on Movement and Computing*, MOCO '22, pages 1–7, New York, NY, USA, June 2022. Association for Computing Machinery.

[19] L. McCallum and M. S. Grierson. Supporting Interactive Machine Learning Approaches to Building Musical Instruments in the Browser. In R. Michon and F. Schroeder, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 271–272, Birmingham, UK, July 2020. Birmingham City University. ISSN: 2220-4806.

[20] G. Meza. Exploring the potential of interactive Machine Learning for Sound Generation: A preliminary study with sound artists. *Proceedings of the International Conference on New Interfaces for Musical Expression*, May 2023. Place: Mexico City, Mexico.

[21] T. Murray-Browne and P. Tigas. Latent mappings: Generating open-ended expressive mappings using variational autoencoders. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Shanghai, China, June 2021.

[22] T. Namgyal, P. Flach, and R. Santos-Rodriguez. MIDI-Draw: Sketching to Control Melody Generation, May 2023. arXiv:2305.11605 [cs, eess].

[23] J. Park. jeonghopark/SketchSynth-Simple, Jan. 2024. https://github.com/jeonghopark/SketchSynth-Simple.

[24] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2360–2368, Red Hook, NY, USA, 2016. Curran Associates Inc. event-place: Barcelona, Spain.

[25] M. Rodger, P. Stapleton, M. van Walstijn, M. Ortiz, and L. S. Pardue. What Makes a Good Musical Instrument? A Matter of Processes, Ecologies and Specificities. In R. Michon and F. Schroeder, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 405–410, Birmingham, UK, July 2020. Birmingham City University. ISSN: 2220-4806.

[26] G. Roma, O. Green, and P. A. Tremblay. Adaptive Mapping of Sound Collections for Data-driven Musical Interfaces. In M. Queiroz and A. X. Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 313–318, Porto Alegre, Brazil, June 2019. UFRGS. ISSN: 2220-4806.

[27] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, Apr. 2015. arXiv:1409.1556 [cs].

[28] T. Tsandilas, C. Letondal, and W. E. Mackay. Musink: composing music through augmented drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 819–828, New York, NY, USA, Apr. 2009. Association for Computing Machinery.

[29] G. Vigliensoni and R. Fiebrink. Steering latent audio models through interactive machine learning. In *In Proceedings of the 14th International Conference on Computational Creativity*, Ontario, Canada, 2023.

[30] R. Yao, C. Liu, L. Zhang, and P. Peng. Unsupervised Anomaly Detection Using Variational Auto-Encoder based Feature Extraction. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7, 2019.

[31] A. Zolotov. Virtual ANS Spectral Synthesizer, 2023. https://warmplace.ru/soft/ans/.

# APPENDIX

## A. COMPOSITION JOURNAL

### A.1 Composition 1

My idea is to build an improvisation performance similar to Luc Ferrari, in which my sketch will trigger different portions of different samples to create a soundscape. I modified a granular synthesiser Max patch and connected it with the macros. The macros can be mapped to the grain position, the gain, and the pitch. I hope the synths can stay silent when the sketch canvas is empty, and trigger sounds when some sketches are presented. I tried two approaches for this: The first one was to pre-process the sample to add a few seconds of silence, and put the grain position to the silent section when the sketch canvas is blank. This didn't work well for long samples like a drum or synth loop because it results in lots of similar patterns when a sample is triggered or stopped. The second approach is to use two macros for one sample, one for gain and the other for grain position.

I started by picking around 10 pieces of samples that sounded interesting. I made 10 copies of the granular synth, put them in separate tracks, and loaded them with the samples. Then, I started recording Wekinator's training examples. I paired short vertical straight lines with two percussive samples by increasing the gain of their synths while drawing short straight lines (the first figure in Figure 7). These two percussive samples are grouped as Group 1. Using a similar approach, I paired vertical wavy lines (the second figure in Figure 7) with an auto-panned piano note sample to create a swirling sound, marked as Group 2. A full drum loop that contains a bass drum, snare, and a few hihats was paired with circles and marked as Group 3. I enjoyed playing with this drum loop, especially when I was sketching fast and some random percussive sounds happened, which made me feel that the sketches were actually "triggering" something. In Group 4, I used a pluck synth loop and paired it with square-like drawings (the fourth figure in Figure 7). For Group 5, all samples used follow a 4-bar chord progression, therefore, I mapped the grain position of these samples to the same macro control to ensure that they are always on the same chord. Group 6 contains two different vocal slices. Mappings used in this composition are listed in Figure 8

However, then I realised that, as I kept adding mappings, it was easier to trigger specific sounds when there were only three to four sets of sounds. But once it went more than that, it became impossible to precisely trigger a specific sound. Therefore, after adding Group 4, I had to program some automation to ensure composability: The composition opens with loosely triggered percussive samples (Group 1). Then, a programmed soft and slow chord progression comes in, followed by Group 2, 3 and 4. Next, I discovered that drawing horizontal lines may occasionally trigger a very high-pitched vocal slice, which I actually don't remember why this happened, so in the performance I tried playing it along with Group 4 (shown in the third figure in Figure 7.

| | | Macro 1 | Macro 2 | Macro 3 | Macro 4 | Macro 5 | Macro 6 | Macro 7 | Macro 8 |
|---|---|---|---|---|---|---|---|---|---|
| Group 1 | Synth 1: Percussion Loop 1 | | P | V | | | | | G |
| | Synth 2: Percussion Loop 2 | | V | | | P | G | | |
| | Synth 3: Foley Loop | | | | | | V, G | P | |
| Group 2 | Synth 4: Piano Note | V | P | G | | | | | |
| Group 3 | Synth 6: Full Drum Loop | G | | P | | V | | | |
| Group 4 | Synth 7: Plucky Synth Loop | | | P | | V | G | | |
| Group 5 | Synth 8: Piano Loop 1 | | G | V | | | | | |
| | Synth 9: Piano Loop 1 | | G | | V | | | | |
| | Synth 10: Piano Loop 1 | | G | | | | | | V |
| | Synth 11: Piano Loop 1 | | G | | | | | V | |
| | Synth 12: Pad Synth Loop | | G | | | | V | | |
| | Synth 14: Bass Loop | | G | | | | | | |
| Group 6 | Synth 13: Vocal Phrase 2 | G | | | | | | | |
| | Synth 5: Vocal Phrase 1 | G | | P | V | | | | |

| |
|---|
| G: Grain Position |
| V: Volume |
| P: Pitch |

**Figure 8: Mappings in the first composition.**

## A.2 Composition 2

For this composition I attempted to map all eight macros to a single wavetable synthesiser. I start by adding potential parameters that could be useful for IML. 13 parameters were selected including filter cutoff, FM index, LFO amount, distortion amount, and a couple of parameters in the effect rack. Then I started to record training examples for the Wekinator model. I created a 4-bar MIDI clip with a sustained chord [D1, D0, F#2] and put it on a loop. I did not have a very clear plan for the composition, so I simply started by arbitrarily tweaking the synthesiser parameters, and then drawing something that I felt matched the sound (some examples are shown in the second row of Figure 7). After about 10 rounds of tweaking and sketching, I didn't quite remember what had been drawn in the first few rounds. Therefore, when I started to play with the mapping, I first tried to reproduce patterns I used during training and see if there were any usable patterns. I also experimented with some shapes that I had never used in training examples to see what would happen, for example, I found that drawing long straight lines (shown in the last figure in Figure 7) produces something sounds like white noise, which is suitable as the ending.

I attempted to avoid using any automation during performance. However, one of the difficulties is that since the mapping is very sensitive to changes in my sketch, I have to slow down the sketching speed to make sure that it doesn't sound like randomly triggered noise. Sometimes it's hard to trigger a specific sound I hope to get, and sometimes I have to wish that it won't accidentally trigger something I'm not expecting.

Beside, I noticed that, when recording Wekinator's training examples, all the parameters are stable and static. However, when the training is done and the system starts running, these parameters are constantly changing and evolving as the sketch is drawn and fades out. Therefore, although I built all the mapping myself, it's still hard to foresee what it would sound like until the system actually starts running.