

Reflections on Eight Years of Instrument Creation with Machine Learning

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

Laetitia Sonami
Center for Contemporary Music
Mills College, Oakland CA US
lsonami@mills.edu

ABSTRACT

Machine learning (ML) has been used to create mappings for digital musical instruments for over twenty-five years, and numerous ML toolkits have been developed for the NIME community. However, little published work has studied how ML has been used in sustained instrument building and performance practices. This paper examines the experiences of instrument builder and performer Laetitia Sonami, who has been using ML to build and refine her *Spring Spyre* instrument since 2012. Using Sonami's current practice as a case study, this paper explores the utility, opportunities, and challenges involved in using ML in practice over many years. This paper also reports the perspective of Rebecca Fiebrink, the creator of the Wekinator ML tool used by Sonami, revealing how her work with Sonami has led to changes to the software and to her teaching. This paper thus contributes a deeper understanding of the value of ML for NIME practitioners, and it can inform design considerations for future ML toolkits as well as NIME pedagogy. Further, it provides new perspectives on familiar NIME conversations about mapping strategies, expressivity, and control, informed by a dedicated practice over many years.

Author Keywords

Machine learning, music composition, instrument design

CCS Concepts

- Applied computing → Arts and humanities → Performing arts;
- Computing methodologies → Machine learning

1. INTRODUCTION

Machine learning (ML) techniques are a core NIME research topic, with “machine learning” appearing explicitly in the conference's call for papers since 2006. Members of today's NIME community have been using ML to create digital musical instruments (DMIs) since at least 1991 [16], and numerous ML toolkits have been created for NIME practitioners over the past decade (e.g., [3],[11],[12],[13],[22]). Yet little published work has examined how ML has been used in sustained instrument building and performance practices within our community. Such reflection is necessary to fully understand the usage and value of ML techniques, which in turn can inform the design of future ML tools, NIME pedagogy, and artistic practice. Further, at a time when the creative use of ML has seen sudden and surging interest among computer scientists (e.g., workshops on Machine Learning for Creativity and Design at NeurIPS [17]), arts organisations (e.g., [1]), and the popular press (e.g., [19]), we believe that NIME practitioners who already have years of practice with ML should contribute to the

wider conversation about the utility and consequences of employing ML in creative work.

This paper therefore closely examines the experiences of instrument builder and performer Laetitia Sonami, who has been using ML to build and refine her *Spring Spyre* instrument since 2012. Prior to making the *Spring Spyre*, Sonami created the *lady's glove* [23] and performed with it regularly from 1991 to 2014. Using Sonami's current practice as a case study, we explore questions such as: What aesthetic and experiential qualities can ML facilitate during the processes of instrument building, composition, and performance? How do the challenges and opportunities presented by the use of ML to build instrument mappings differ from those present when building instruments using programming alone, without ML? How, specifically, can ML be employed to build a sustainable instrument with sufficient flexibility and complexity to support continued engagement over a period of many years?

This paper additionally provides a first-hand account from Wekinator creator Rebecca Fiebrink, whose collaborative design work with Sonami has informed numerous refinements to the Wekinator software. The paper concludes with a discussion connecting our first-hand accounts to each other and to concerns of the broader NIME community. We reflect on topics such as the utility of ML for NIME, the design of mapping and instrument-building tools, open challenges for improving NIME pedagogy and practices with ML, and the value of engaging in cross-disciplinary collaborations and sharing the knowledge gained from them.

2. BACKGROUND

ML has been used to create new musical instruments and interfaces since at least the early 1990s, when Lee, Freed and Wessel employed neural networks to control sound synthesis [16]. Around that time, Fels and Hinton [6] also performed early experiments using neural networks to control speech synthesis using a sensor glove.

Since then, NIME researchers have created numerous toolkits to facilitate the use of ML in instrument building, particularly to support parameter mapping—i.e., to define the functions that compute synthesis control parameter values from gestural or other input values, as described in [14]. Examples of general-purpose ML tools for music include Wekinator [11], Gesture Recognition Toolkit [13], and ml.lib [3]. The above tools employ a supervised learning approach—using classification, regression, and/or dynamic time warping algorithms—to facilitate mapping from performer actions to music or sound control parameters. Typically, an instrument designer begins by creating a set of training examples consisting of example performer actions (e.g., positions or motions sensed with sensors) paired with example sounds (e.g., values of synthesis parameters). A learning algorithm is trained on these examples, which builds a model of the relationship between performer actions and music or sound parameter values. This model can be used in live performance, where it computes new parameter values in response to a performer's current actions. Typically, such tools also support an “interactive machine learning” [5] approach, in which instrument creators can quickly and iteratively refine models by



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

providing new examples, for instance to correct a model’s mistakes or introduce new sounds.

Recent research on the use of ML in instrument design has developed further variations on this standard supervised learning framework in order to account for the specific design needs of instrument builders and gestural interaction designers. For instance, Françoise et al. [12] designed the XMM toolkit to support creators designing temporal and multimodal gesture-sound mappings. Scurto et al. [22] developed and evaluated deep reinforcement learning techniques that aim to aid sound designers in exploring sound parameter spaces. Scurto and Fiebrink [21] developed a “grab-and-play” mapping method for very quickly creating a mapping that accommodates a performer’s physical abilities or preferences, and Parke-Wolfe, Scurto, and Fiebrink [18] deployed it in an ML tool for music therapists and teachers working with children with disabilities.

Despite the frequency with which new ML systems or approaches appear in the NIME literature, the literature has few examples of research interrogating the experiences of people using such systems over long periods of time. Yet work that has engaged deeply with users has led to surprising insights. For instance, Fiebrink and collaborators [7][10] used several months of participatory design workshops and interviews with composers employing an early version of Wekinator to develop an initial understanding of how ML may be useful in instrument design. They found, for instance, that ML was useful in reducing barriers to design exploration, privileging the gesture-sound relationship rather than requiring composers to think about their work in terms of the programming code or underlying mathematics, and inviting play [10]. Parke-Wolfe, Scurto, and Fiebrink [18] discovered that a music teacher and music therapist who worked with ML over a period of nearly a year enjoyed using ML-based mapping generation not only because it made instrument design accessible to non-programmers, but also because it enabled them to create interactions that supported children in exercising agency, encouraged movement and listening, and supported social aims such as increased participation in group music-making. Yet to our knowledge, no published work has investigated how composers or performers have engaged with ML over longer periods. Do the benefits to the design process suggested by the preliminary work above (e.g., [10]) persist? What other benefits and challenges arise when ML becomes a key part of one’s practice? What characteristics of ML tools may help or hinder such practice? How do the answers to these questions suggest we may support students learning about the potential value and practicalities of using ML in their work?

3. MOTIVATION AND METHOD

As an instrument builder/composer/performer using ML in music-making (Sonami) and a researcher/software developer creating the ML tool for this music-making (Fiebrink), we believe our experiences with ML over the past eight years are of value in addressing the questions above. A pioneer in our community, Sonami has been building and performing with new digital instruments since 1991, when she created the *lady’s glove* [23] (see also description in [2]). The *lady’s glove* is an elegant, elbow-length black Lycra glove capable of sensing a variety of characteristics of hand shape and motion. Over the past three decades, Sonami has frequently performed with the *lady’s glove* at international venues such as Ars Electronica, the Interlink Festival, and NIME 2014 (as a keynote speaker/performer). Since 2012, Sonami has been using Wekinator [11] to build and perform with a new instrument, the *Spring Spyre* (Figure 1). Sonami has performed her own work with *Spring Spyre* in over 50 performances, as well as performing work by composer Eliane Radigue. These performances include solo works as well as improvisation with other musicians. Sonami has a unique insight into the value and challenges of ML; we know of no other creator who has used ML so extensively in instrument design or performance.

Fiebrink is the creator of the Wekinator software used by Sonami, and she has designed and taught creative machine learning classes at



Figure 1: The *Spring Spyre*

Goldsmiths University of London and online on the Kadenze platform [15]. As described in Section 5, her discussions with Sonami have driven a number of design changes to the software over the past eight years, and they have also informed her teaching practice.

Our aim in the following sections is to reveal aspects of our experiences with ML that are of interest to the broader NIME community, as motivated in the previous section. Given the personal and specific nature of these experiences, a conventional academic approach to describing them (i.e., an ostensibly objective, dispassionate, third-person account) seems inappropriate. We believe that first-hand accounts of our experiences can more appropriately reflect their subjectivity, their complexity, and their grounding in our own practices and goals. Therefore, we present Sections 4 and 5 below as first-hand accounts written independently in response to a set of questions, derived using the following process: We frequently hold informal discussions about our practice during self-generated residencies, and through these discussions we identified a set of issues we felt were most relevant and interesting to the NIME community. We then collaboratively generated and refined two sets of interview questions that engaged with these issues—one set for Sonami and one for Fiebrink—and we each independently responded in writing to our assigned questions. Sections 4 and 5 contain the final interview questions as well as each respondent’s answers in her own words, with light collaborative edits for clarity. In Section 6, we return to a more conventional academic tone for a discussion informed by our reflections on the interview responses and their connections to some broader concerns of the NIME community.

4. SONAMI IN HER OWN WORDS

4.1 The *Spring Spyre*

Q: What is the history of the *Spring Spyre*? And what was your motivation for making it?

A: David Wessel and I experimented with machine learning and the *lady’s glove* in the mid 90’s. The training was done with Michael Lee’s software running on Max, and the synthesis models on the SGI. While exciting, the system was complex and not suitable for easy manipulation and touring. I stayed with my intricate web of mapping (without ML) till meeting Rebecca Fiebrink at Princeton in 2010. Rebecca demonstrated the early stages of her Wekinator. Impressed by its ease of use and flexibility, I decided to build a new instrument which would be based on this platform. I could have implemented ML with the *lady’s glove*, but the mapping I had elaborated in Max over twenty years, while unwieldy, was intrinsically linked to the conception of pieces performed with the *lady’s glove*. As for the *lady’s glove*, it had become a very fine instrument but it was time to move on.

Wanting to start from a blank slate and generate a new approach and new imagination, just a few of the elements from the *lady's glove* were carried over to the *Spring Spyre*. I wanted to improve the likelihood of unpredictable events, which I learnt to cherish in the *lady's glove* when they occurred. I wanted to retain some interdependence of inputs (in the *lady's glove*, one muscle of one finger, when moved, will affect other muscles in other fingers).

Aside from these, I was looking for more complex inputs and opted for a partially chaotic system which would “fight” the intention of ML and not learn (!). I ultimately used thin springs attached to audio pickups. These would allow for movement of the springs to continue after having been activated by my hands, as opposed to the paradigm of the *lady's glove* in which hand movement is tightly mapped to the sound: less theatricality and yet more mystery, the audience (and myself) having somehow lessened our expectation/anxiety of a correlation between gestures and sounds. Or maybe I had grown tired of justifying the gestures... Everything else relied on the discovery of this new form of mapping and its musical implications.

Q: What is the current implementation of *Spring Spyre*—its hardware, software, synthesis, and use of ML?

A: The current design, which I have now experimented with for eight years, comprises of three thin springs attached to three audio pickups (hacked from cheap reverb tanks). These are anchored to a metal wheel found in a surplus store which itself is anchored to a modified Roland PC1600 controller. This older controller provides sturdiness as well as sixteen decent faders and buttons to mix the various synthesis and control the feature extraction. While I originally intended to attach the pick-ups and springs to any structure I could find, I settled for the past three years on one design so to focus on the instrument.

The audio signals generated when touching the springs are sent to Max/MSP for feature extraction (Figure 2). The audio from each spring is analyzed using 5 biquad filters following 5 prescribed frequencies which can be adjusted based on the spring setup (those though rarely vary as I use the same setup). This provides 15 control values—the instantaneous signal amplitude of each filter output, sampled at a rate specified in the Max/MSP patch, for each of the three springs. I also use the sum of amplitude for each spring's filters, for a total of 18 control values. (The biquads turned out to be much more efficient and flexible than the FFT I had started with.)

These 18 values are sent to Wekinator as inputs [i.e., “features”] to drive its machine learning models [specifically, a set of multilayer perceptron neural networks, one network per synthesis parameter]. Wekinator uses these to control various synthesis methods in real time.

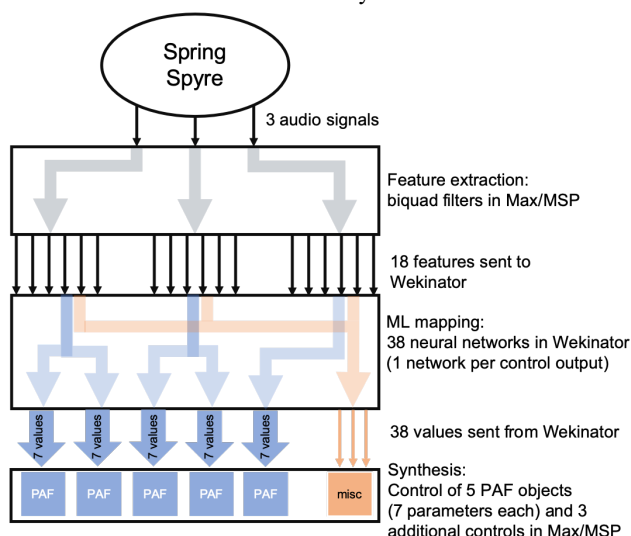


Figure 2: Example of a typical architecture for a *Spring Spyre* piece (synthesis, mapping, and signal routing may change between pieces)

Currently most of the synthesis is done with Miller Puckette's phase-aligned formant (PAF) synthesis algorithm [20]. Different pieces use different numbers of PAF objects as well as other synthesis methods. Rather than having all three springs control all synthesis objects, features from each spring or a combination of springs can be assigned to individual synthesis objects (Figure 2). This kind of flexibility is important.

Q: How do you play the *Spring Spyre*? What kind of music do you make with it?

A: Now that the structure of the instrument is temporarily settled on, most of the work comes in refining the models for the control of the synthesis. While I may keep some models from piece to piece, I do introduce new ones or keep modifying previous ones. This is the core of the composition and requires a lot of patience and attention. While ML allows for fast experiments in mapping which is invaluable, the work resides in refining the synthesis so it will respond in rich and unexpected ways. The springs are “live;” they oscillate in various ways and slowly die. How does the synthesis respond to these various behaviors?

The unpredictability I referred to earlier depends on how “wide” the machine learning is. If I feed the system training examples whose sounds encompass wide changes based on how I touch the springs, the trained models will move through all these points in unpredictable ways as the springs settle to a resting place. If I give it training examples with narrower changes, the sound will just oscillate slightly as I move the springs. I can thus easily scale the instrument between predictable and unpredictable results by changing how I train. I refer to these variations as the “synthesis terrain”, a nod to David Wessel's “Timbre space”. This “predictability index” is very easily modified and unique to ML.

One interesting use I discovered is that I can train the system with wide points, and when the sounds move through interesting synthesis terrains in performance, I can manually “freeze” the values of some of the inputs that Max/MSP sends to Wekinator, for instance updating just three or four features per spring, thus honing in on a particular aspect of the synthesis. My gestures then have a narrower and more subtle effect on the sound. I can also freeze all the input values if I want to stay on a particular state. I use the buttons on the PC1600 to do this. This forces me to listen very actively during the performance so I can “catch” the sounds. This active listening is challenging, exciting and is new. This is the main difference in the actual live performance using the *Spring Spyre*. While the *lady's glove* required a very focused attention to keep track of how my gestures would affect the thirty sensors attached to the arm, the mapping would be fixed as to how the sounds would be affected by the gestures. This is a very exciting part of ML: the ability to move across synthesis terrains, discover new sounds, and refine the control in live performance.

4.2 ML as Creation Tool

Q: How do you think about machine learning as a tool? Do you think of it essentially as a mapping tool, or do you think of it as something with independent intelligence or agency?

A: I wish I could say I approach ML as having independent intelligence or agency, but I essentially use it as a mapping technique and it is part of a system. It cannot be dissociated from the hardware (the springs and pickups) and the software (Max/MSP). These three components define the instrument. The instrument does have agency and identity.

This is why I can say that it took eight years to get to a point when I started to understand “what the instrument wants”. Only recently has it become more of an “exchange” between the instrument and me, the performer. Not just forcing my intentions onto it, but letting it inform the composition and performance. Still much work remains in my explorations with ML and its application to musical performance. It is an open system which allows for continuous exploration in sound synthesis, expressivity and adaptability.

Q: Can you say more about how you think about mapping in your instruments and compositions? How does ML impact on the task of creating a mapping?

A: Mapping is the backbone of a composition. While it has been argued that in electronic instruments, controllers, sound generators and links (mappings) are independent units, I believe those to be tightly correlated. The choice of control inputs dictates the gestures, hence possible mappings and resulting musical events. While people using more standard controllers such as faders and buttons might apply control as an afterthought, mapping strategies have been an essential component in elaborating pieces with the *lady's glove* and now with the *Spring Spyre*. Mapping bridges the physical world to the sonic world. ML offers a way to easily configure mappings with a wide variety of behaviors, thus allowing the composer to focus on the sounds and compositions. I cannot envision at this point in time any other ways to map.

Q: Do you use ML as a tool to build instruments, pieces, or both? How do you think about these when using ML?

A: Pieces are defined by the instrument. If you were to write for the piano you would not compose for the flute. Pieces for the *lady's glove* could only be performed with the *lady's glove*. Pieces for the *Spring Spyre* can only be performed with the *Spring Spyre*. I am not interested in the universality of the instrument.

Each piece varies, based on the models the ML was trained on. ML makes it easy. I may go through hundreds of models before settling to a particular “palette” for a particular piece.

The performance of the piece can be well defined in advance (as is the case of the piece composed with Eliane Radigue, OCCAM IX), or more improvisatory if the models used are expansive, as explained earlier.

Q: What are some of the challenges you encounter in working with ML?

A: The main challenge is the lack of available synthesis methods which could take advantage of the ability to control large numbers of parameters in real time. For years limitations were dictated by the availability of affordable A/Ds and interfaces. ML can now allow many input controls to be targeted to many synthesis parameters. You can for instance have 80 inputs, all modified in real time, control the synthesis. But what synthesis currently allows for such dynamic control? People have suggested physical models, but models are fragile, and easily break when controls go off range. We need more research to allow for complex synthesis now. This is very frustrating.

I also encounter practical challenges working with Wekinator. For instance, one current challenge is the inability to bundle trained models and export them to different pieces [the software currently only supports exporting and loading individual models, one at a time]. Each spring for instance may control one synthesis engine, thus 6 inputs and 8 outputs, trained many numbers of times. I'd like to be able to bundle them, call them one “instrument”, and move them to another piece to supply the initial space of inputs and outputs.

Also, as I've started teaching Wekinator for the design of instruments and control systems, I still encounter what I consider misunderstandings. The focus on applying ML to precise, independent, and predictable control of individual parameters in prepared compositions is prevalent. I think most often musicians apply control and mapping strategies after they have composed their pieces and resist opening up their compositions to allow these to interfere, influence or even hijack their original plan. This seems to be a poor application of ML but the desire for control still prevails over the desire to explore new forms of expressivity.

5. FIEBRINK IN HER OWN WORDS

Q: Describe how you have worked with Laetitia over the past eight years.

A: I gave Laetitia a demo of the first version of Wekinator in 2010. I was delighted when, a short time later, she expressed an interest in using ML to build a new instrument and shared with me some of the

physical prototypes of what would become the *Spring Spyre*. I observed her experiment with a variety of physical configurations and feature extraction methods until she landed on the audio pickups with biquad filters, which seemed to get her enough information to train some very interesting models. Since then, we try to spend time together at least a couple of times a year, with long walks discussing our work and hacking sessions trying out new ideas. During these, we've had innumerable discussions about how she is using Wekinator (both the practicalities of its use and the aesthetic aims of her work), what she has found frustrating or confusing in the software, and what new features she would like to see implemented. When I redesigned Wekinator from the ground up in 2013, I started by showing her paper prototypes of the new user interface I envisioned, and she was the first user to try it out. Since then, many of the new features I've implemented have been driven by her ideas.

Q: What have you learned from working with Laetitia? What has most surprised you?

A: I've learned to see ML less and less as something whose value for instrument building lies primarily in its ability to build accurate models from training examples (the way a computer scientist would conventionally see it), and more as a technique that is wonderful for supporting richer modes of interaction with computers during both performance and instrument building. The composers in my first Wekinator participatory design workshops in 2009 planted the seeds of this way of thinking (e.g., see [10]), but my discussions with Laetitia over the last eight years have really cemented my understanding of how ML can be used to create complex mappings that couldn't be built using programming. And this has pushed me to design tools that better support that type of use in practice.

I was initially very surprised by Laetitia's conviction around making an instrument that didn't prioritise control in a conventional sense. *Spring Spyre* is not an instrument that you play using a specific, practiced set of gestures in order to achieve a precise sequence of sounds. Yet this doesn't mean she wants complete chaos, or to be surprised all the time. There is a complex balance at play, in which Laetitia precisely specifies some things—such as which springs drive which synthesis modules—and somewhat less precisely specifies others—such as the width or scope of these synthesis terrains. Together, these yield an interface that allows freedom to explore spaces that are new and surprising, while also likely to be engaging and relevant.

Because of this, I now think about “mapping” in instrument design as the process of creating a new interactive world—ideally one that invites exploration and discovery; this is quite different from the way it is sometimes discussed in the NIME literature as primarily a set of engineering choices that together impact on controllability, learnability, ergonomics, etc. (e.g., see [14]). By extension, I believe that many of the criticisms of the notion of “mapping” (e.g., Chadabe's [4] view that mappings are one-directional, simplistic, and control-focused) have limited applicability, as mapping can alternatively become a dynamic and deeply engaging activity, in which the computer becomes a creative partner of sorts. I've written previously [8] about how tools for instrument design that support such activity can be viewed as “meta-instruments,” and my work with Laetitia strongly informed that writing.

Q: What changes have you made to Wekinator in response to your collaborations with Laetitia?

A: I began this line of research thinking that my observations of composers working with Wekinator would suggest new types of ML algorithms to better support them. However, the truth is that the biggest payoffs for improving composers' experiences have come from rethinking the design of the user interfaces around the ML. In response to Laetitia's needs for greater flexibility, control, and ease of use of ML, I've made a large number of changes to the software in the last eight years. I've made it easier to define and change feature selection architectures, for instance specifying which springs impact on which

synthesis modules. I've added a lot of infrastructure to support long periods of evolution, experimentation, and archiving of creative projects, for instance making it easier to save and load trained models and to share models between projects (though as Laetitia has noted, this could still be improved by supporting "bundling" of models to reflect how a composer may think about groups of models as conceptual units). I've made it easy to "undo" and "redo" changes to models' training, to support more flexible and low-cost exploration of different mappings. I've also made it easier to reduce the CPU load by turning unused models "off" until they're needed, and by minimizing GUI CPU use.

The first version of Wekinator allowed OSC communication with feature extractors and sound synthesis modules in any language, but Wekinator was built partly with Chuck [24] and it privileged Chuck integrations: it included several built-in feature extractors in Chuck (for audio and HID devices) and provided Chuck programmers with a light-weight API for implementing new synthesis modules. This made it very easy for new users to experiment with audio and HID inputs, and made it simpler for Chuck programmers to connect to Wekinator, but at the cost of making the whole program more conceptually complex for everyone. I simplified Wekinator 2.0 (released in 2015) by removing native Chuck components so that all feature extractors and synthesis modules had to communicate using OSC in the same way. I then worked with Laetitia to design a new API for OSC control of Wekinator, which now provides simple but powerful ways to integrate Wekinator with work in Max or other environments that can send and receive OSC. This not only allows full control over the ML process from the same environment in which synthesis is happening, but it also supports more complex musical structures: for instance, a composer can transition to a new section of a piece (with a new mapping) by sending Wekinator an OSC message triggering the loading of a specific new set of trained models.

While many of these changes may be uninteresting from a research point of view, they're driven by my realization that, at the end of the day, I want to support work in which ML isn't "the point" of an instrument. ML cannot be merely a gimmick if someone is going to use it in their work for years; it has to actually be useful. And ensuring a creator can use ML seamlessly (or at least relatively painlessly) within their creative process is paramount, even though it can take a lot of time to build ML tools that do this well.

It's also important to mention that I would have never understood the need for all these specific changes to Wekinator based on my own intuition, or using observations of casual users in a workshop setting. It's only by observing the software's use by Laetitia and other intensive users (such as my university students using Wekinator to implement complex music and art projects over the length of a term or longer) that I've been able to identify and prioritise such improvements.

Q: How has your work with Laetitia impacted how you teach interface design and creative machine learning?

A: I've now taught over 100 undergraduate and masters students in term-long creative machine learning courses, and I've had several thousand students in my online course [15]. The biggest impact of working with Laetitia has been on my choices about what to teach them. The set of knowledge that musicians and artists need to do work like Laetitia's is very different from what a typical computer science ML course is likely to cover. Rather than deep knowledge of how algorithms work or how to compare different models' accuracy, for instance, students need to be able to reason about how changes to features and training examples are likely to change a model's behavior. They need to be able to reason about how they might use such actions to "debug" a trained model when it is not giving them useful music or interaction possibilities. They need to be able to design effective architectures—good feature extractors and feature processing (e.g., normalization), choices about which features influence which models, and choices about how model outputs influence music or artistic parameters.

Furthermore, I try to expose students to the full scope of how ML can be useful in creative practice. Similar to Laetitia's experience, I find it is common for students to initially want to design interfaces that yield precise and simplistic control (e.g., I wave my hand one way and trigger one sound; I wave it another way and trigger a different sound). While ML can support such interfaces (and can sometimes make them easier to build than programming, especially if analysis of video or many sensors is involved), such interfaces do not often enable new types of music nor new modes of engagement between musicians and sound. It is sometimes challenging to convince students to consider that other types of interaction may be worth exploring, but having toolkits that make such exploration easier does help.

6. DISCUSSION

Sonami's account in Section 4 reveals several ways in which ML can be useful in long-term music and composition practices. Some of these align with and expand on previous findings from shorter engagements with users. For instance, work by Fiebrink et al. in 2010 [10] suggested that providing access to surprise and discovery in mapping creation, and that enabling creation of more complex mappings, were valuable aspects of ML compared to designing mappings using programming alone. Sonami's answers above reveal how she has managed to sustain these benefits in her work over many years: this includes the use of a physically complex input system, feature extractors that succinctly capture relevant information from this system (and which can be adapted to different configurations of the input system), and synthesis methods that provide access to a very wide variety of sounds under different parameterizations.

Further, Sonami has developed working strategies to better enable her to create a wide variety of pieces and performance techniques with the same physical setup. This includes manipulating the "predictability index" of a given mapping through her strategy for training the ML, intervening in the ML process in novel ways to achieve particular musical results (e.g., temporarily "freezing" feature values), and managing complexity within and across pieces by designing modular signal flows (e.g., creating mappings that control individual synthesis algorithms with a single spring, allowing modular mapping components to be shared across pieces or swapped in and out over sections of a piece).

To support such a practice, ML tools must support flexible, low-level control over a number of aspects of ML. As described above, Sonami's practice has required Wekinator to provide control not only over the iterative re-training typical of interactive machine learning approaches, but also over precise feature selection and signal routing, as well as the saving, loading, and swapping of ML components during composition and performance time. These actions clearly support creative exploration during composition and the performance of more musically complex works, but such actions are not typically foregrounded in more general-purpose ML APIs or GUI-based tools (which often assume that users' efforts will be more focused on tuning and comparing different learning algorithms).

Sonami's practice prioritises exploration, discovery, and navigating varying degrees of predictability and control. In such a practice, the act of creating a mapping is one of crafting sound and interaction spaces, rather than designing a control paradigm. This suggests new dimensions that might be considered in the design and evaluation of new instruments and instrument creation tools: for instance, do they enable effective manipulation of a "predictability index," or make it easy to discover engaging new "synthesis terrains"? We are excited to see work in this vein by Scurto et al. [22], which explores algorithmic and interactive techniques that are explicitly designed to support exploration of new sounds.

Sonami's approach to using ML has implications for teaching, as discussed by Fiebrink: if students are to use ML in similar ways to build instruments, this demands particular knowledge about ML practices, which is quite different from what is covered in a typical ML class in a computer science department. Further, both of us have

frequently experienced difficulty encouraging students to move beyond simplistic control paradigms in their work building new creative interfaces. Fiebrink has advocated above for exposing students to tools that make it easier for students to explore alternative paradigms; while Wekinator is one such tool, one can envision many other approaches, including simpler tools for making “many-to-many” (see [14]) mappings without ML.

Sonami has noted the paucity of synthesis methods that seem able to take full advantage of the type of mappings offered by ML, in which a potentially huge number of control parameters are simultaneously adjusted, often using non-linear and many-to-many mappings, providing access to a very wide space of potential sounds and sound trajectories. We believe it is exciting to consider what type of new synthesis algorithms might be constructed to exploit usage in mappings created with ML.

Finally, we note that just as there is little published literature exploring how composers or performers have engaged with ML or other NIME technologies over a period of many years, there is little literature revealing how NIME technology creators have worked with and been influenced by their users over such time periods. Fiebrink has emphasized above the value of working with Sonami and observing other users over time; these have led to new ways of thinking about the value of her tools and to many tool improvements. We hope to see other such creators engage in long-term collaborations or longitudinal study of users’ experiences, and to see published work sharing how insights from such collaborations influence their technology design.

7. CONCLUSION

In this paper, Laetitia Sonami has described her experiences using ML as an instrument builder, composer, and performer. These experiences illuminate some of the benefits, challenges, and working strategies that may arise with the use of ML in instrument mapping creation over a sustained period. Sonami’s discussion of her practice also reveals new perspectives on familiar NIME conversations about mapping and control. Rebecca Fiebrink has additionally described here how her work with Sonami has changed her understanding of mappings and of the utility of ML in instrument design, and led to concrete changes to the Wekinator ML software and to her teaching. We have discussed how these experiences can deepen understanding of the value of ML for instrument creation, and how they can inform NIME research, tool development, and pedagogy. Many of our insights in this domain have been made possible through our long-term collaboration as an artist and a technology creator, and we urge others in the NIME community to engage in such collaborations as well as to find ways to reflect on and share the knowledge emerging from them.

8. ETHICAL STANDARDS

This work is unfunded. We are not aware of any potential conflicts of interest.

9. REFERENCES

- [1] Barbican. 2019. AI: More than human. Exhibit at the Barbican, London UK, 16 May–26 Aug 2019. <https://www.barbican.org.uk/whats-on/2019/event/ai-more-than-human>
- [2] B. Bongers. 2000. Physical interfaces in the electronic arts. In M. M. Wanderley and M. Battier, eds., *Trends in Gestural Control of Music*. IRCAM—Centre Pompidou, pp. 41–70.
- [3] J. Bullock and A. Momeni. 2015. MLlib: Robust, cross-platform, open-source machine learning for Max and Pure Data. In *Proc. NIME*, 265–270.
- [4] J. Chadabe. 2002. The limitations of mapping as a structural descriptive in electronic instruments. In *Proc. NIME*.
- [5] J. A. Fails and D. R. Olsen, Jr. 2003. Interactive machine learning. In *Proc. International Conference on Intelligent User Interfaces (IUI '03)*, 39–45.
- [6] S. S. Fels and G. E. Hinton. 1995. Glove-Talk II: An adaptive gesture-to-formant interface. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 456–463.
- [7] R. Fiebrink. 2011. Real-time human interaction with supervised learning algorithms for music composition and performance. PhD thesis, Princeton University, USA.
- [8] R. Fiebrink. 2017. Machine learning as meta-instrument: Human-machine partnerships shaping expressive instrumental creation. In T. Bovermann et al., eds., *Musical Instruments in the 21st Century*, 137–151. Springer.
- [9] R. Fiebrink. 2019. Machine learning education for artists, musicians, and other creative practitioners.” *ACM Transactions on Computing Education*. Vol 19, No. 4, September 2019.
- [10] R. Fiebrink, D. Trueman, C. Britt, M. Nagai, K. Kaczmarek, M. Early, M. R. Daniel, A. Hege, and P. R. Cook. 2010. Toward understanding human-computer interaction in composing the instrument. In *Proc. of the International Computer Music Conference (ICMC)*.
- [11] R. Fiebrink, D. Trueman, and P. R. Cook. 2009. A meta-instrument for interactive, on-the-fly machine learning. In *Proc. NIME 2009*.
- [12] J. Françoise, N. Schnell, R. Borghesi, and F. Bevilacqua. 2014. Probabilistic models for designing motion and sound relationships. In *Proc. NIME 2014*.
- [13] N. Gillian and J. A. Paradiso. 2014. The Gesture Recognition Toolkit. *The Journal of Machine Learning Research*, 15 (1), 3483–3487.
- [14] A. Hunt and M. M. Wanderley. 2002. Mapping performer parameters to synthesis engines. *Organised Sound*, 7 (2), 97–108.
- [15] Kadenze, Inc. Machine learning for musicians and artists. Online course taught by Rebecca Fiebrink. <https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists/info>
- [16] M. Lee, A. Freed, and D. Wessel. 1991. Real-time neural network processing of gestural and acoustic signals. In *Proc. International Computer Music Conference (ICMC)*, 277–280.
- [17] Machine learning for creativity and design. Workshop held at the 2019 NeurIPS conference. <https://neurips2019creativity.github.io/>
- [18] S. T. Parke-Wolfe, H. Scurto, and R. Fiebrink. 2019. Sound Control: Supporting custom musical interface design for children with disabilities. In *Proc. NIME*.
- [19] S. Paul-Choudhury. 2019. What will music be like in 20 years? BBC Culture, 21 May 2019. <http://www.bbc.com/culture/story/20190521-what-will-music-be-like-in-20-years>
- [20] M. Puckette. 1995. Formant-based audio synthesis using nonlinear distortion. *Journal of the Audio Engineering Society*, 43 (1/2), 40–47.
- [21] H. Scurto and R. Fiebrink. 2016. Grab-and-play mapping: Creative machine learning approaches for musical inclusion and exploration. In *Proc. International Computer Music Conference (ICMC)*.
- [22] H. Scurto, B. Van Kerrebroeck, B. Caramiaux, and F. Bevilacqua. 2019. Designing deep reinforcement learning for human parameter exploration. arXiv:1907.00824.
- [23] L. Sonami. “Lady’s glove.” <http://sonami.net/ladys-glove/>
- [24] G. Wang and P. R. Cook. 2003. Chuck: A concurrent, on-the-fly audio programming language. *Proc. International Computer Music Conference (ICMC)*.