Deep Predictive Models in Interactive Music

Charles P. Martin* Kai Olav Ellefsen† Jim Torresen‡

Abstract

Musical performance requires prediction to operate instruments, to perform in groups and to improvise. We argue, with reference to a number of digital music instruments (DMIs), including two of our own, that predictive machine learning models can help interactive systems to understand their temporal context and ensemble behaviour. We also discuss how recent advances in deep learning highlight the role of prediction in DMIs, by allowing data-driven predictive models with a long memory of past states.

We advocate for predictive musical interaction, where a predictive model is embedded in a musical interface, assisting users by predicting unknown states of musical processes. We propose a framework for characterising prediction as relating to the instrumental sound, ongoing musical process, or between members of an ensemble. Our framework shows that different musical interface design configurations lead to different types of prediction. We show that our framework accommodates deep generative models, as well as models for predicting gestural states, or other high-level musical information. We apply our framework to examples from our recent work and the literature, and discuss the benefits and challenges revealed by these systems as well as musical use-cases where prediction is a necessary component.

1 Introduction

Prediction is a well-known aspect of cognition. Humans use predictions constantly in our everyday actions [17], from the very short-term, such as predicting how far to raise our feet to climb steps, to complex situations such as predicting how to avoid collisions in a busy street and, finally, to long-term planning. Prediction can be defined as guessing unknown or future states of the world informed by our current and past experiences. When our predictions are not accurate, such as lifting our feet for one too many

*University of Oslo, Department of Informatics, `charlepm@ifi.uio.no`, ORCID: 0000-0001-5683-7529

†University of Oslo, Department of Informatics, `kaiolae@ifi.uio.no`, ORCID: 0000-0003-2466-2319

‡University of Oslo, Department of Informatics, `jimtoer@ifi.uio.no`, ORCID: 0000-0003-0556-0288

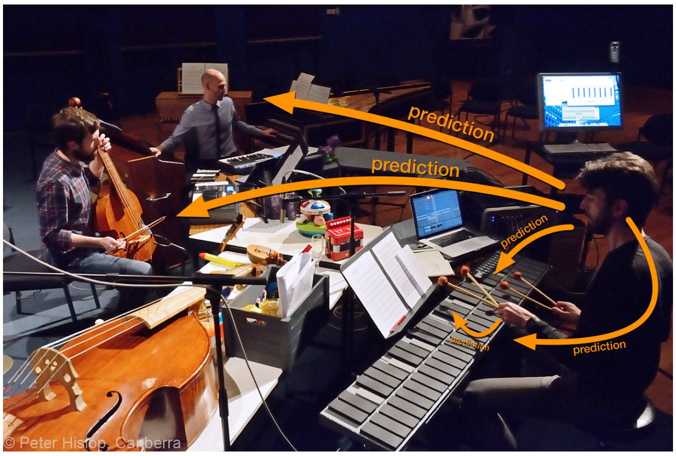

Figure 1: Many types of cognitive prediction are required in musical performance from low-level instrumental control to high-level planning using multiple senses. Musical machine learning models can be used in digital instruments to support performers' predictions. (Photo: © Peter Hislop)

steps, the error is used as a warning to correct our actions; in that case, the warning is the sensation of surprise. Neuroscientists are now able to observe prediction in action in the human brain. In particular, prediction has been observed for visual perception [62], as well as musical perception [65]. Other researchers have theorised that prediction and expectations are key to our aesthetic appreciations [11], and, indeed, that prediction is the fundamental basis for intelligence [36].

Musical performance involves many layers of prediction (see Figure 1). Skilled performers predict the sounds produced by different instrumental gestures; they predict the musical effect of rehearsed expressions and improvised sounds; and they predict the musical actions of an ensemble. It may seem natural that interactive music systems and digital musical instruments (DMIs) should incorporate prediction to better account for the complexity of musical performance. Brown and Gifford have noted that prediction has been only modestly implemented in such systems [11], particularly for incorporating proactivity into musical agents.

In contrast, we feel that many DMIs already use predictive models of various kinds. These models are often used to generate new musical data, manage ensemble experiences, or handle complex sensor input. Unfortunately, the design frameworks that are often called upon to understand these DMIs do not generally consider the role of prediction; they tend to focus on *reactive* rather than *predictive* operation.

In this paper we investigate how DMI designs using predictive models can lead to new creative affordances for performers and DMI designers. We introduce a new framework for understanding predictive models in

DMI designs in terms of their relationship to the cognitive predictions that musicians use to perform. We show how a number of existing DMIs have applied predictive models to supplement these cognitive predictions, extending and supporting the performer's creativity. These systems apply various machine learning (ML) and artificial intelligence (AI) approaches; however, we review where recent work in deep learning has had particularly meaningful applications in DMIs and where it could be used in future systems.

A practical contribution of our framework is that it provides a way to frame two important, but usually separate, problems in computer music—mapping and modelling—as different sorts of predictions. Mapping refers to connecting the control and sensing components of a musical instrument to parameters in the sound synthesis component [39]. While acoustic instruments often have no separation between the control mechanism and sound source (e.g., a guitar string), the separation in electronic instruments allows the potential for many exciting and creative mappings, but also design difficulties. Modelling refers to capturing a representation of a musical process [23]. The model can be used to generate further music [2], or help understand music that has been created. Both of these problems have heuristic, as well as ML approaches. While mapping is one of the main problems in interactive music system design, modelling is more focussed on non-real-time composition as a main application.

Mapping and modelling have parallels in the musical performance predictions shown in Figure 1. Performers learn to predict the sonic result of their control gestures; this could be called "instrument prediction" which involves building a cognitive mapping between control and sonic output. Performers also do higher level "musical prediction" of the notes or gestures they play either by looking ahead in a score, or planning and selecting from different musical possibilities in an improvisation. Musical prediction clearly involves modelling musical processes at various levels. Finally, in a group situation, performers predict the action to sound relations and high-level musical direction of other musicians or a conductor. This involves both mappings and high-level models learned through experience.

By rethinking mapping and modelling as different kinds of predictions, we can bring multiple musical applications of ML under one framework. This exposes some future opportunities for endowing DMIs with predictive intelligence. It also helps to understand some of the challenges of predictive DMIs, such as interacting in ensemble situations, and handling temporal effects such as rhythmic, harmonic and melodic structures.

In the next section we will discuss what prediction can mean in a musical context and review the development of musical deep learning models that can potentially be applied in interactive music design. In Section 3, we will provide a framework for incorporating these predictive models into DMIs. In Section 4, we will examine how this framework can be applied to two of our interactive music systems and several systems from the literature. Finally, in

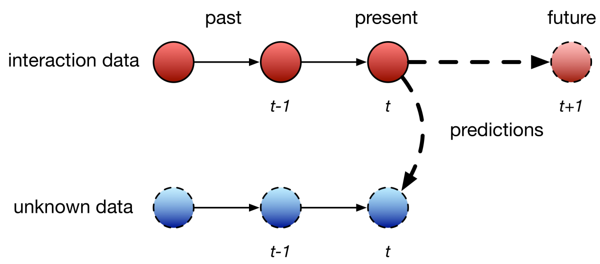

Figure 2: Predictive models in an interactive context can predict future states of a known sequence or the present state of an unknown process.

Section 5, we will outline the benefits and challenges that predictive models can bring to DMI designers and performers.

2 Prediction and Music

Cognition involves many levels of prediction that we rely on for our everyday actions [17]; however, it is not always clear how prediction could be integrated into creative tools in a beneficial way. In this section, we will discuss what prediction can mean in an interactive system, and what musical predictive models show most promise for interactive use.

2.1 What is a Prediction?

A simple definition for prediction could be as follows: the estimation of unknown data based on current knowledge and perceptions. This definition encompasses the everyday understanding that prediction relates to data in the future (e.g., weather predictions), as well as the ML understanding of prediction as simply any unknown variable (e.g., image classification). In an interactive music application, perceptions will generally consist of sensed information about the performer and musical environment. Knowledge will consist of previous experiences summarized in a learned model and latent variables. Unknown data can come in two main varieties, as shown in Figure 2: future values of the sensed information, or some different process running in parallel.

For future predictions, the sensed information may include the performer's movements or gestures, symbolic musical data, or high-level information about the musical context. In ML, this kind of temporal prediction is often referred to as sequence learning [76] or time series forecasting [16]. Predictions do not have to relate to the future. In ML, the two typical types of prediction tasks are classification and regression, where models are trained to predict categorical and quantitative data respectively [35]. Both of these terms are often applied when predicting a different type of data than that given as input, without supposing any temporal relationship. Such present predictions can have a role in musical interaction as well; for instance, a model might predict classifications of musical technique from gestural sensors.

2.2 Models of Musical Sequences

Using automatic systems to generate music is a compelling and enigmatic idea. From the rules of counterpoint and music theory, to explorations of indeterminacy in musical composition and performance by composers such as John Cage or Iannis Xenakis, algorithmic composition has been practiced for centuries. More recently, artificial neural networks (ANNs) have been used to generate musical compositions and, now, digital audio signals directly. Recurrent neural networks (RNNs) are often used to generate sequences of musical notes in a one-by-one manner, where the input is the previous note and output is the next predicted note to occur. Mozer's CONCERT system [56] is an early example of this idea. The later introduction of gated units such as the long short-term memory (LSTM) cell [38] improved the ability of such networks to learn distant dependencies. RNNs with LSTM cells were later used by Eck and Schmidhuber to generate blues music [24]. These models have a flexible ability to learn about the temporal context in a sequence and thus mimic human cognitive abilities for sequence learning and prediction [18].

Other popular systems for generating music use Markov models to generate the emission probabilities of future notes based on those preceding [3, 23]. The advantage of RNN models over Markov systems is the latter requires unreasonably large transition tables to learn distant dependencies in the data [56]. RNNs can make more "fuzzy" predictions, interpolating between the training examples, rather than attempting to match them exactly [31].

The proliferation of GPU computation and large datasets has contributed to the popularity of creative RNN models. Character-level text generation [42], is now well known in computational arts. Music, too, can be represented as text and generated by an RNN such as the "ABC" formatted folk songs of the FolkRNN project [75]. More complex musical forms such as polyphonic chorales of J. S. Bach have also been modelled by RNNs; Hadjeres et al's work on DeepBach allows such a model to be steered towards generating voices to accompany certain melodies [33]. RNN models can even be combined with the rules of music theory via a reinforcement learning tuning step described by Jaques et al. [40]. Google's Magenta project[1] has developed a collection of RNN models for music generation and has notably released trained versions of several musical RNN models and used them in creative tools and experimental interfaces.

[1]Magenta - Make Music and Art Using Machine Learning: https://magenta.tensorflow.org.

These models learn much about the temporal structure of music, and how melodies and harmonies can be constructed; however, there is more to music than these aspects. Sturm et al. [74] acknowledge as much, calling the output of FolkRNN "transcriptions" of (potential) folk tunes, not tunes themselves. These transcriptions have a melody, but musicians need to contribute their own arrangement and expression to perform them as complete musical works.

Some recent models have begun to integrate more aspects of music into their output, and thus produce more complete performances. Malik and Ek's StyleNet [47] annotates existing musical scores with dynamic (volume) markings. Simon and Oore's PerformanceRNN [70] goes further by generating dynamics and rhythmic expression, or rubato, simultaneously with polyphonic music. In terms of representations of music, PerformanceRNN's output could be said to be *thicker* [21] than FolkRNN's thin output, because it contains much more of the kind of information required to actually perform a musical work.

Of course, an even thicker representation of music would be the actual sounds of the performance. WaveNet models [79] can render raw audio samples using dilated causal convolutional layers, rather than a recurrent network, to handle temporal dependencies. These models are capable of producing samples, the short musical sounds that can be used in music production [25], as well as translating between different "styles" of music [55]. These models show great promise; however, computational requirements have not been sufficiently overcome for them to be widely explored in an interactive context.

3 A Framework for Predictive Musical Interaction

In this section, we argue that a very simple, three-stage framework of interactive music systems can be extended to include models for different types of prediction. This framework divides interactive music systems into three stages: sensing, processing, and response, as shown in Figure 3, originally due to Rowe [66]. While this framework is simple, it provides a helpful division of concerns and has previously been used to frame DMI designs [27, 22]. One benefit of this framework is that it demonstrates that electronic music systems, unlike most acoustic instruments, are modular. The sensing and response stages in particular are often interchangeable, for instance, different interface designs (e.g., keyboard, wind, or percussion controllers) could be used with the same synthesiser.

In the context of research-focussed musical interfaces, the elements of this system are often prototype electronic systems, and bespoke software. A simple experimental DMI might, for instance, collect data from rotary potentiometers at the sensing stage, map this data to control of synthesis parameters in a computer music environment such as Pure Data running on

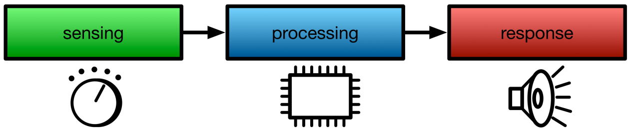

Figure 3: The three-stage model for interactive music systems.

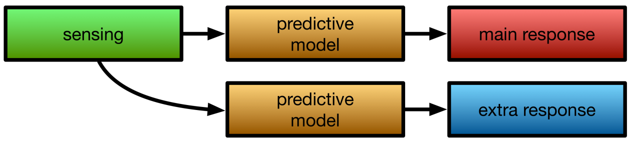

Figure 4: In instrument prediction, a model replaces the processing stage of a DMI, or acts as a parallel mapping to an extra response stage.

a Raspberry Pi in the processing stage, and the response could be audio that is sent to a speaker system. Such an arrangement is illustrated in Figure 3. More complex DMIs, or systems that span across an ensemble, may have multiple sensors, processing stages, and responses.

A predictive model can be considered as an extra component of this framework, providing extra, or unknown, information in some part of the DMI. Predictive models have some flexibility about the type of information they have as input and output and can connect to this framework at various points. Input could come from either the sensing or response stage, and output could be directed to any stage: sensing, as some additional generative sensed data; processing, as parameters or adjustments to the mapping; or response, as commands for a synthesis system.

We characterise the *type* of prediction in these models as one of three important types of cognitive prediction that could occur during human music performance, highlighted in Figure 1: *instrument* prediction, *musical* prediction, and *ensemble* prediction. In each of these cases, the predictive model supplements existing cognitive predictions that a human performer uses unconsciously.

Human musicians learn some kind of model that relates the movements or gestures for controlling an instrument with the pitch, duration, volume, and timbral quality of the resulting sound. An instrument prediction model does the same thing, and so is highly connected to the concept of mapping that was discussed above.

Figure 4 shows how an instrument predictive model can replace the “processing” stage of a DMI, by serving as a mapping between possible sensor values and synthesiser parameters. The predictive model could also connect to a separate response in parallel, for instance a visualisation or extra instrumental sound. In general, this allows complex mappings to the response stage to be created from learnable mappings to sensor inputs. This

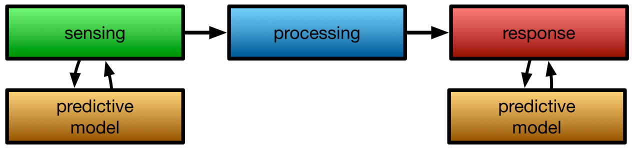

Figure 5: In musical prediction, a model generates future musical possibilities, either at the sensing stage (where predictions could be control data), or response stage (where predictions could be future notes or sounds).

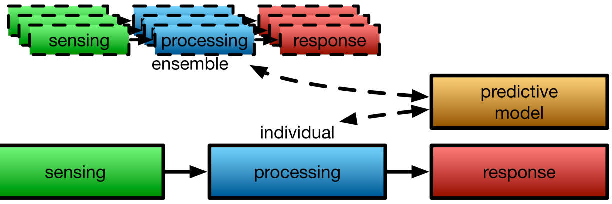

Figure 6: In ensemble prediction, a model predicts information about other members of an ensemble, this could be musical notes or other salient events.

process has been previously understood as a design tool for creating DMIs [27], but can be considered as a supplement to cognitive prediction that reduces the cognitive load on human musicians, or extends their creative reach.

While performing, musicians rely on a cognitive model of future musical possibilities, either by reading ahead in a score, following a learned piece, or improvising new music. A musical prediction model can allow a DMI to also generate such predictions and act on them sonically. In Figure 5, we consider where a musical prediction model could fit into the DMI framework. The model could take sensor data as input—in this position it would be predicting the performer's control gestures. A predictive model at the response stage would predict high-level musical outputs, for instance symbolic music. The output of the predictive model could be fed back into the system at the same stage, resulting in an instrument that "plays itself" in some way.

Interacting with other musicians in an ensemble requires many different predictions; for instance, anticipating the rhythmic pulse of an ensemble, that one musician will play a solo, or the best dynamic to enhance the collective sound. In this paper, we simplify these to a general concept of ensemble prediction, where a model takes data from any other member of a musical group and sends output to a local performer's DMI (see Figure 6). Ensemble predictions could include musical predictions, such as for generating "virtual" ensemble members to accompany a solo performer, as well as instrument prediction for tasks such as early transmission of note signals to networked musicians to account for latency.

4 Predictive Interactive Music Designs

In this section, we review a number of DMI designs that include predictive models, including some examples of our own work (see Table 1 for a complete listing). These examples are arranged by the type of prediction occurring, and we discuss the configuration and purpose of the predictive models in the DMI design and the ML the models used. While many of these existing systems do not use deep learning models, they show how predictive interaction can be incorporated into creative tools and artistic practices.

4.1 Instrument Prediction

The potential of ML models to predict the parameters of sound synthesis systems from gestural or control input has been acknowledged since at least the early 1990s [44]. One early application was Fels and Hinton's GloveTalk II system [26], where a number of ANNs connected hand and finger sensors to a voice synthesiser. This system was trained to produce vocal sounds from examples given by the user. Rather than designing a mapping and demanding that the user learns their own predictive model from gesture to sound, ML models applied in instrument prediction adapt to gestures the user already expects to use, supplementing the user's cognitive predictions.

4.1.1 ML as mapping

Many artists and researchers wish to connect complex or multiple sensors to the parameter controls of audio or computer graphics systems. As with GloveTalk II, this can often be accomplished effectively with classical ML models such as shallow ANNs or k-nearest neighbour classification [1]. Artists have been aided in this regard by frameworks such as Wekinator [29], that embed such algorithms into interactive music environments, allowing them to be trained interactively by recording examples of control data matched to the expected output classes or parameters. In practice, training such models on-the-fly and iteratively allows for valuable creative exploration of their affordances and predictive power [28, 27].

Snyder's Birl [73] is a series of self-contained electronic wind instruments where continuous-valued buttons (e.g., capacitive sensors) are used as the control input. One iteration of the Birl used an ANN to map between these buttons and the pitch of the synthesised sound. This ANN was trained interactively using Wekinator, but later implemented on a microcontroller. The advantage of the ANN over a hand-built mapping in this case was that designed fingering-to-note mappings could easily be learned, but the ANN also interpolates between these fingerings (i.e., when a button is not fully touched) and creates some, perhaps unpredictable, output for untrained combinations. This ML approach, however, is potentially more difficult to understand than rule-based or physical model approaches to mapping that were also used with the Birl [72].

The use of predictive models in the processing stage is becoming more common in interactive music designs; however, these models do not always consider the temporal component of the data. As a result, they may not be able to model all aspects of the musical interaction. For instance, if a sensor can measure hand position, a non-temporal model might be able to map the position of the hand to a response, but not the direction of the hand's motion. Using RNNs, rather than non-recurrent ANNs for instrument prediction could better account for temporal effects in performance.

The above listed systems have all used supervised learning to generate algorithms for instrument prediction, with sets of training data provided either by a DMI designer or performer. An interesting alternative is applied in the Self-Supervising Machine [71]. In this system, real-valued input data is segmented during performance by an adaptive resonance theory algorithm [15], and these examples are stored to progressively train and re-train a shallow ANN mapping to synthesis parameters. Among several use-cases, the model is used with input data sourced from a touchscreen, and from the sonic features of a violin. This system allows all learning to take place with an interactive musical performance session; however, as the predictive model is unknown until it is created, the performer needs to learn their own model of the DMI's behaviour without practice, as the authors note, this "lack of constraints can be challenging" [71].

4.1.2 Predicting extra responses

Many DMI designs seek to augment existing musical instruments with audio effects, extra sounds, or visual elements. When performers literally have their hands full, a predictive model may be able to interpret gestural information from cameras and other sensors to control these extra responses.

In 000000Swan's Monster [68], Wekinator's predictive models are used to track the output from a Kinect camera and a K-Bow, a sensor-laden bow for string instruments [54]. Output from these models are used to control triggering of audio samples, parameters on audio effects, and computer generated visuals. The performers provided training examples by matching demonstrations of sensor input with desired synthesis and visual configurations in Wekinator.

The PiaF or Piano Follower [78] is an augmented piano system designed to track auxiliary gestures in the pianist's hands during performances and use these to control synthesised sounds including processing of the piano audio. The core of the system consists of a piano keyboard connected to an audio processing system with sound output. A Kinect depth-sensitive camera captures the position of the performer's hands, arms, and body during the performance which is sent to a gesture variation follower (GVF) algorithm

[14]. This temporal ML model tracks multiple dimensions of input data to classify from a number of trained gestures. GVF additionally provides continuous data about the speed, scale and rotation of the gesture. This is particularly useful in a creative interface where important expressive control, for example over timbre in a musical instrument, could be encoded in control variations.

When operating PiaF, the performer's movements throughout a composed performance are broken down into a sequence of gestures during a training phase. During performances, data from the Kinect is sent to the GVF system to determine which gesture is being performed (and thus, which part of the performance is being played). This, and variation data about that gesture are used to control parameters in the audio processing part of the system. The result is a system that can enhance the pianist's expressive options during performance.

In Monster and PiAF, the output of the predictive models were directly tied to parameters in synthesiser and visualisation systems; however, ML models can also be directed to more abstract, high-level, classifications. The BRAAHMS system uses a functional near-infrared spectroscopy (fNIRS) headset to measure the "cognitive workload" of a piano performer [80] and a support vector machine (SVM) classifier. This system adds generative harmonic lines to melodies playing on the piano under either low- or high-workload conditions. Ben-Asher and Leider used a naive Bayes approach on pianists' hand movements to classify the emotional content of their performance into six high-level categories [6]. These classifications were used to drive a visualisation during performances. Deep ANN models might be able to predict high-level information, such as ratings of expression or rhythmic accuracy, directly from audio signals [61].

4.2 Musical Prediction

ML models for musical prediction build a representation of the notes, sounds, or other playing instructions in a musical work or corpus. Such models are often directed towards capturing a certain musical style [23] and generally are configured to future notes on the basis of those previously played. Musical models can be used in two main ways: predicting future notes, which can be played back or compared to those actually played, or used to analyse the music that has already been played [19]. In the following examples, musical prediction is generally used to fill-in musical parts that the performer doesn't play, or to continue when they stop.

4.2.1 "Continuing" Musical Interactions

The *Continuator* is a DMI that models and imitates the style of individual performers to "continue" their performances where they leave off [59]. The performer plays on a control interface where high-level MIDI note data is the output to be sent to a synthesis module; this MIDI data is also tracked by the Continuator. As soon as the performer stops playing, the Continuator activates, generating new MIDI notes in the same style as the performer's recent input and sending them directly to the synthesiser. When the performer resumes playing, the Continuator ceases the imitation and goes back to tracking their performance. The temporal predictions here are generated by a variable order Markov model that chooses from the space of various notes and rhythms entered by the performer. This relatively simple model allows the system to learn on-the-fly but limits the range of temporal dependencies that can be represented.

Beatback is an interactive musical model focussed on drum machine performance [37]. Similarly to the Continuator, Beatback uses a variable order Markov model which is trained during performance from musical material supplied by the performer. In drum machine patterns, performers play notes on the different sounds of the acoustic drumset: bass drum, snare, hi-hats, etc. Beatback's call-and-response mode predicts likely continuations of the user's complete drum pattern when they stop playing. A second "accompaniment" mode functions differently, by only predicting notes for instruments that the user leaves out. For drum machine performance, and unlike many other instruments, predicting simultaneous musical phrases can serve as a practical augmentation for solo performance, rather than a duet.

Deep RNN models can be applied to musical continuation in a similar way to the Markov models. The Magenta project's AI Duet [48] integrates their Melody RNN model into an interactive music system that can run as part of a computer music environment or in a self-contained web application. The Melody RNN attempts to predict new notes from those in the recent past—it automatically activates during performance, playing back its predictions in a different voice allowing the user to engage in call-and-response style improvisation. Where the Continuator's Markov model was trained on the performer's own playing, the Melody RNN needs to be pre-trained on a large corpus of MIDI data. In practice, the ability to learn from a very large corpus of data can be a significant advantage; a novice user might provide very simple musical input and the RNN could encourage or inspire them with more elaborate musical ideas.

The systems above operate on symbolic music as both input and output but it is also possible to produce continuations of audio signals. OMax is a system of agent-based musical prediction models designed for use in interactive improvisation [4]. This system can handle polyphonic MIDI data as well as audio, so can be used by musicians playing acoustic instruments. OMax allows one, or multiple, factor oracle models [5] to be trained in real-time during a musical performance from streams of symbolic music data. The system can capture audio signals, use pitch-tracking or some other feature analysis to classify the signal into sequences of classes that are used to train the predictive models, and then respond in the performer's own sound using concatenative synthesis of the recorded signals [4]. In the future, audio feature analysis in a similar system could be handled by a deep belief network (DBN) [34] or convolutional neural network (CNN) (e.g., [8]), which could be trained offline on larger amounts of audio data.

4.2.2 MySong

MySong is a system to automatically generate harmony accompaniments for vocal melodies [69]. The predictive model takes as input a vocal melody sung by the user and outputs a sequence of chords that match the melody. The melody and chords can then be played back together allowing the user to hear a piano arrangement of their performance. The predictive model in MySong blends predictions made by a hidden Markov model (HMM) and a simple, non-temporal model of chord probability based on the notes that appear in each musical measure. The user is able to tune the predictions to emphasise the HMM or melodic chord assignment, as well as a parameter between models learned from songs divided between major and minor modes.

The benefit of MySong's predictive model is that a user is able to hear their vocal improvisations in the context of a full musical arrangement, a much more complete musical work. MySong supports the user's creativity and allows them to reflect more productively on their performances by predicting an appropriate harmonic context. Although MySong plays back a piano accompaniment to the melody, we categorise this model as making musical predictions as the chords relate mostly to the melody rather than the response of another performer. MySong's HMM model seems to have been effective, but recent research suggests that bidirectional RNNs can achieve better predictions with more diverse, and perhaps more interesting, chord sequences [46].

4.2.3 RoboJam

RoboJam [53] is a call-and-response agent developed by the authors that uses an RNN to continue musical performances created in a smartphone app [52]. RoboJam is unique in using this RNN to model musical control data rather than musical notes. In this way, the predictive model sits at the sensing stage of the interactive music framework.

In this application, performers using a smartphone app collaborate asynchronously by contributing 5-second performances to a cloud-based music system. The short performances are created by simple mappings of touch-screen taps and swiping to notes played by various synthesiser instruments. RoboJam conditions an RNN on these short performances and predicts an additional 5 seconds of control data. This predicted data is used to play a different synthesiser and layered with the original performance. These

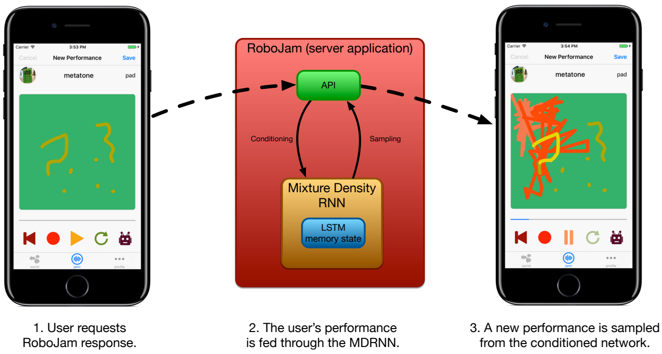

Figure 7: RoboJam is a call-and-response agent for continuing touchscreen musical performances. It uses an RNN to generate a sequence of real-valued touch interactions after being conditioned on a user's performance.

are musical (rather than ensemble) predictions as they continue the performer's own control data. This system allows users to hear more complex performances quickly, and to hear their performance in context with different layered sounds.

RoboJam's predictive model is trained on a corpus of musical touchscreen interactions which consist of touch locations and the time since the previous interaction. The model uses a mixture density RNN inspired by models of line drawings [32] to predict sequences of this real-valued data. Importantly, this model is able to predict the rhythm of interactions absolutely, rather than quantised to a set number of steps per measure.

Since the model predicts low-level musical control data, rather than notes, it could be said to learn how to *perform* music, than how to *compose*. This arrangement means that RoboJam has access to the whole expressive space of the touchscreen mapping and can potentially perform very convincing responses. Since RoboJam learns to play through the touchscreen, its performances can also be played through any of the synthesis mappings available in the app; so, if the user performs using a string sound, the RoboJam response might be played back with a drum sound. While low-level learning has benefits, it comes at a cost of difficulties in training—RoboJam's continuations are yet to be as musically convincing as AI Duet's.

4.3 Ensemble Prediction

In ensemble prediction, a model predicts the actions of other members of a musical group. In this section we review a number of systems in two broad categories: those that use such models to support networked performance scenarios; and those that simulate an ensemble from the performance of a single musician. We also discuss an example of our own work where an RNN is used to simulate a free-improvisation touchscreen ensemble.

4.3.1 Network Ensemble Prediction

In network musical performance, groups of musicians perform together over network connections from different physical locations [43]. Time delays over networks are unavoidable and can prevent convincing performance depending on the physical distance, system latency, and the temporal sensitivity of the music. Predictive models between the musicians can allow information to be transmitted ahead of actual musical events, allowing the music at each end to be correctly synchronised. These predictive models tend to be used for instrument prediction, as the models often aim to create a predicted version of the musician's response stage at the remote end of a network connection.

The MalLo system accomplishes this task for percussion performances by incorporating a predictive model into a percussion instrument [41]. This model, described by Oda et al. [57], uses computer vision techniques to track the position of the percussionist's mallets and quadratic regression to predict when the mallet will strike the instrument before this actually occurs. By predicting mallet strikes, MalLo can preemptively send note data to remote participants which is scheduled to occur in time with the local sound. MalLo's model applies temporal instrument prediction to support ensemble performance. Similar systems have been implemented to predict Indian percussion patterns [67], and to support massed ensemble performances using a common metronome [13].

In a related form of ensemble prediction, the collective behaviour of a group is collected over a network and analysed to identify important events in a performance. In Metatone Classifier [50], control data from a touchscreen ensemble is sent to a central server that, first, uses a Random Forest classifier to identify high-level gestures and, secondly, generates a Markov model to predict whether the ensemble has collectively changed its style of improvisation. This information is sent back to the touchscreen DMIs to trigger changes in the individual interfaces.

4.3.2 Simulated Ensemble Prediction

Individual musicians often engage in simulated ensemble experiences of different kinds, from practice and performance with a fixed backing track to the popular use of looping effects. With predictive models, these experiences can be made reactive and flexible to the changing behaviour of the performer. These applications usually include some kind of musical prediction, to model the behaviour of other musicians in an ensemble and to understand the performance of the live soloist to provide appropriate accompaniment.

A relatively well-explored form of simulated ensemble prediction is score following, where an algorithm tracks a performer's progression through a known musical score to provide a synthesised accompaniment synchronised to the soloist [20]. The task of tracking the performer's location is often accomplished with a hidden Markov model where the performer's notes are the observed states and score locations are the hidden states [58]. The Orchestra in a Box system uses an HMM in this way and provides accompaniment by playing back a time-stretched backing track [63].

For styles of music such as jazz, rock, and pop, a "thick" musical score is usually not available, and so more advanced predictive models are needed to create the accompaniment. In many cases, these can be constructed using a combination of rule-based and ML systems. Biles' *GenJam* system [7], for instance, uses genetic algorithms to generate appropriate jazz-style accompaniments with fitness determined from the rules of music theory. This system is also able to engage in interactive improvisation with the human performer by mutating their improvisations to create responses.

In some cases, accompaniments can be generated from a musician's own musical material. The Reflexive Looper [60] records, manipulates and plays back audio from the musician to create an accompaniment. Unlike a simple live looping effect that allows a musician to record a loop and subsequent layers (typically using a pedal interface), the Reflexive Looper uses predictive models to automatically choose audio material to play and manipulates it to a known harmonic progression. A support vector machine is used to classify the performer's recent notes as either melodic, chordal or bass playing, a generative music system then chooses appropriate backing recordings from the two classes that are not being played. While the sound material was generated and manipulated in real-time, the structure of the performance in the Reflexive Looper was limited to pre-determined chord progressions and song structures [49].

The above systems generally represent a "virtual" ensemble only through sound or simple visualisation, although these musicians can also be embodied as robots playing physical acoustic and electronic instruments [10]. For example, the marimba-playing robot Shimon has been used in various interactive music scenarios [9], and employs predictive models for tracking human musicians, prediction of musical notes to play, and communication through physical gestures. Robotic music systems require other types of prediction to control physical movements, a focus of the SHEILA system for imitating drum patterns [77], but these are beyond the scope of this paper.

4.3.3 The Neural Touchscreen Ensemble

The Neural Touchscreen Ensemble [51], a system developed by the authors, is an RNN-driven simulation of a touchscreen ensemble experience. A human performer plays freely improvised music on a touchscreen and an ensemble performance is continually played back on three RNN-controlled touchscreen devices in response. Both the human and computer-controlled devices use a simple app that allows struck or sustained sounds from a limited selection of notes. The performer's touch control data is sent to the

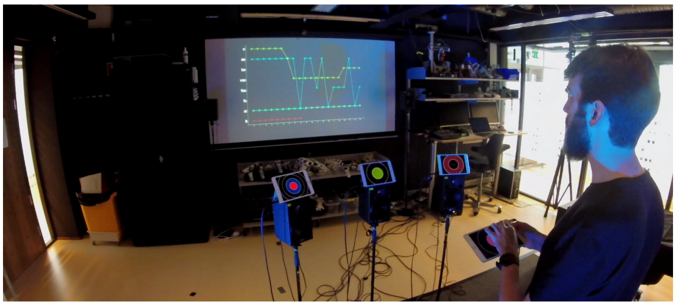

Figure 8: The Neural Touchscreen Ensemble uses an RNN to predict ensemble responses to a human performer's gestures. The system supports quartet performances with three RNN-controlled iPads responding to one human performer.

server which, using a Random Forest classifier, predicts a high-level gestural class for the latest data once every second. The classes come from a simple vocabulary of 9 touchscreen gestures described in previous research on iPad ensemble performance [50]. The RNN uses three layers of LSTM units and is configured to predict gestural classes in three parallel sequences, that of the three ensemble performers. Four gestures are taken as input—the human performer's present gesture, and the ensemble's gesture at the last time step—and the RNN outputs the three gestures for the ensemble at the present step. Control signals matching the gestures can then be played back by the ensemble devices from a corpus of performance recordings.

This system uses both an instrumental prediction model (touchscreen control data to gestural class) and musical prediction at the ensemble level to predict potential responses. The RNN ensemble model is temporal as it uses previous experience stored in the LSTM units' state to make predictions. Although the whole system takes touchscreen control data from the sensing stage as both input and output, the RNN model predicts only high-level gestures. Control data is generated from these gestures by a touch synthesis module. In a more advanced system, touchscreen interactions could be directly predicted as in RoboJam. The Neural Touchscreen Ensemble's musical content—freely improvised touch interaction—would not be easily described by music theory used in GenJam or the Reflexive Looper above. A data-driven approach to modelling this kind of interaction was required.

5 Conclusion

In this paper, we have defined a framework that connects predictive models in interactive music systems with cognitive predictions involved in performing music. We have reviewed how existing systems, including two from our own group (RoboJam and the Neural Touchscreen Ensemble), have implemented

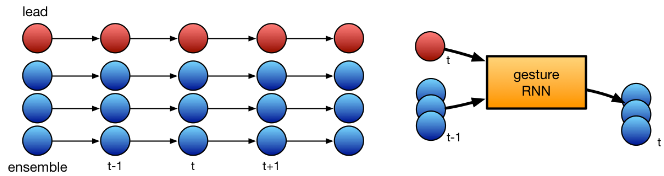

Figure 9: Gesture-RNN predicts appropriate gestural motions for three ensemble members based on present information about the human performer, and past information about the ensemble.

predictive models. This framework enhances our understanding of how ML models can support performers and enhance their creative potential. A variety of ML techniques have been employed, including models that forecast future values of a known time-series, or that predict the present value of an unknown quantity. In each case, predicting this unknown data has allowed the systems to do more than we would normally expect of a musical instrument. They are able to act preemptively, to make more expressive use of the user's musical control data, and to predict ensemble responses from artificial agents or remote participants.

Our review demonstrates that deep learning models, in particular, have much to offer predictive musical interaction. RNN models can learn from large corpora of training data allowing wide musical experience to be included in a DMI. This contrasts with Markov-based musical prediction models that tend to learn only from the performer's contribution. Deep models are also extremely flexible and can be designed to predict multiple dimensions of related data simultaneously with the same temporal model. We took advantage of this ability in both of our systems. In RoboJam, we were able to predict touchscreen interactions in both 2D space and absolute time, a novel improvement on typical step-based musical models. The Neural Touchscreen Ensemble uses a typical RNN design, but the input and output one-hot vectors actually encode multiple performer gestures. Despite the interest in deep ANNs for generating symbolic music, few interactive music systems apply these as predictive models. We suggest that other musical deep models could be incorporated into interactive music designs to take advantage of their flexible capacity for data-driven prediction, and potential to generate low-level output such as control or audio data.

Although we have discussed many interactive music systems that use ML models, these are not often characterised in relation to cognitive prediction. We think that this undersells the importance of prediction in these systems and in musical performance in general. Embedding predictive intelligence into DMIs appears to be a crucial step towards creating interfaces that allow more expression, follow performers more naturally, and engage more closely with ensembles and audience. Our framework could be used to help implement these models in musical systems. In the final part of this paper we will discuss what we see as the benefits that predictive models can offer to DMI designers and performers, as well as some challenges that they may face.

5.1 Benefits of Prediction

Music and sound are *temporal* phenomena and yet, the widespread framework for interactive music systems shown in Figure 3 does not necessarily consider the axis of time. Indeed, the fundamental archetype for DMIs is reactive; response necessarily follows gesture. We think that predictive models are vitally important to embedding a temporal axis into interactive music design. In reality, predictive models *are* used in DMI design. Considering prediction as an essential part of interactive music design frameworks allows these temporal models to be properly understood, examined, and developed. This issue has gained increasing relevance in recent years due to deep learning models enabling new insights into the difficult problem of predicting long-term structure in music [64].

While traditional acoustic musical instruments are (necessarily) reactive, their players are not. Musicians are constantly *proactive* whether anticipating a conductor's beat or introducing a musical idea in a free jam. By embedding predictive models into DMIs, instruments can be proactive as well, to the benefit of their users and listeners. Indeed, in situations where reactive design is insufficient for successful performance, such as networked ensemble performance, predictive systems such as MalLo have been successful. We envisage that proactive elements could be deployed much more widely in DMIs; interfaces could change to afford upcoming musical needs as well as respond to the users' commands.

Typical interactive music designs often include many configuration parameters in the processing stage of their architecture. Predictive models can be used to *adapt* these parameters to meet musical requirements of the performer, audience, or composer. In the PiaF system, we have observed that the GVF model adapts audio processing parameters according to the speed and size of predicted gestures. Indeed, predictive adaptations in an interactive music system could go much further than processing parameters. Virtual reality, touchscreen, or haptic interfaces could be designed to adapt their complexity or functionality according to a predicted requirement.

One of the clearest use-cases for predictive models in interactive music design is to *generate* musical data that reflects the recent style of the user. Automatic music generation, however, can sometimes seem like a solution in search of a problem (Who wants to listen to AI generated music when you can play it yourself?). Both our RoboJam and Neural Touchscreen Ensemble systems use predictive generation to enhance solo performances. In RoboJam, response performances are generated so that the user can hear their own work in context, while in the Neural Touchscreen Ensemble, the actions of three RNN-controlled musicians are generated and synthesised in real-time during the performance.

A strong motivation to continue the introduction of deep generative models into DMIs is that the musical data of new interfaces is often unknown and not well-modelled by music theory. Predictive RNN models, such as that used in RoboJam, could be able to learn a wide variety of low-level control data. Future DMIs could even use deep models with digital audio data as input or output. These could replace multiple parts of the three-stage DMI framework and provide multiple types of prediction simultaneously.

5.2 Limitations and Challenges

Adding predictive models to DMIs can present many challenges to designers and performers. From a design perspective, it can be challenging to develop and train ML models that are *artistically stimulating*. Environments that allow classical ML models to be trained in near real-time (e.g., [29, 12, 30, 45]) assist DMI creators to experiment and evaluate the creative potential of these models [28]. Similarly responsive environments for deep models are yet to appear, although the Magenta project has made moves in this direction. As a result, the integration of RNNs and other deep models into DMIs has been limited.

Where deep models are applied, they can present further difficulties. Models that represent lower level data, such as the control signals in RoboJam, tend to be more *difficult to train* than symbolic music predictors. This is partly due to larger amounts of training required in comparison to Markov systems, or shallow ANNs and is an ongoing challenge in our research.

In all predictive models, the *predictions are limited to the knowledge available in the training data.* The neural touchscreen band's RNN is trained on performance segments and not whole performances. As a result, it can be difficult to get the simulated ensemble to start playing, and to stop at the end of performance. This shortcoming suggests that a few rule-based elements could be helpful, even in data-driven models.

Understanding predictions in a DMI can be challenging for performers. For instrument prediction, this is sometimes overcome by including the performer in the training process [68, 26]. For models that are trained during performance, the performer needs to continually update their own understanding of the model in parallel, which can become overwhelming. DMIs that change their mappings under a musician's fingers run the risk of frustrating rather than engaging the performer. Musical prediction models that simply continue when the performer is not playing allow the performer time to listen and understand. For systems where extra sounds are generated by a predictive model in synchrony with the musician, the source of these sounds must be clear. One strategy is to follow structured performance paradigms such as jazz interaction, or live-looping; another is to physically embody these responses in robots or visually represented instruments.

5.3 Final Remarks

Prediction has clear roles in musical performance. In this work we have shown how predictive models can fit into DMI design by complementing and extending the cognitive prediction already used by performers. Our review has used this framework to explore the musical and creative consequences of different types of prediction. In a world where AI and deep learning interactions are increasingly built into everyday devices, the place of predictive models in musical interaction certainly bears scrutiny. While DMI designs show strong use of multi-modal sensing, highly creative processing, and artistically savvy responses, predictive models have sometimes been under-explored. We argue that considering machine learning models in DMIs as extensions of human cognitive predictions helps to explain their benefits to users and performers. Deep models, such as RNNs, are being widely explored for music modelling, but, despite their flexibility in learning large and low-level musical sequences, are not yet widely used in DMIs. Future deep predictive models may be able to handle multiple types of prediction in a DMI, with end-to-end mappings from sensors directly to sound. To achieve these deep predictive DMIs, we challenge musical interface designers to consider prediction as a new framework for ML in interactive music.

5.4 Funding

This work was supported by The Research Council of Norway as a part of the Engineering Predictability with Embodied Cognition (EPEC) project, under grant agreement 240862.

5.5 Conflict of Interest

The authors declare that they have no conflict of interest.

Table 1: Predictive interactive music systems reviewed in this paper ordered by type: instrument (inst.), musical (music.), and ensemble (ens.). The machine-learning model used, and its input and output configuration are also listed.

Title	**Description**	**Model**	**Input**	**Output**	**Type**
GloveTalk II [26]	speech synthesis control	MLP	hand sensors	synthesis parameters	inst.
PiaF [78]	control effects with hand gestures	GVF	Kinect	audio effect parameters	inst.
BRAAHMS [80]	adaptive harmonisation using BCI	SVM	fNIRS	harmonisation parameters	inst.
Emotionally intelligent piano [6]	control visualised colours	naive Bayes	IMU	emotion classes	inst.
The Birl [73]	control synth through button interface	MLP	capacitive button sensors	continuous pitch	inst.
Self-Supervising Machine [71]	control synth through unsupervised control interface	ART, MLP	touchscreen control data	synthesis parameters	inst.
000000Swan's Monster [68]	control extra synth and video layers	MLP, classifier	Kinect, k-bow	synthesis and video parameters	inst.
AI Duet [48]	continue performance	RNN	symbolic music	symbolic music	music.
RoboJam [53]	continue performance	RNN	touchscreen control data	touchscreen control data	music.
Continuator [59]	continue performing in user style	Markov	symbolic music	symbolic music	music.

Table 1: Predictive interactive music systems reviewed in this paper ordered by type: instrument (inst.), musical (music.), and ensemble (ens.). The machine-learning model used, and its input and output configuration are also listed.

Title	**Description**	**Model**	**Input**	**Output**	**Type**
MySong [69]	automatic accompaniment generation for vocal melodies	HMM	audio	chord sequence	music.
OMax [4]	improvisation agent	Markov, FO	symbolic music	symbolic music	music.
Beatback [37]	continue or fill-in drum performances	Markov	symbolic music	symbolic music	music.
GenJam [7]	band accompaniment in real-time	GA	symbolic music	symbolic music	ens.
MalLo [41]	predict percussion strokes	quad. regr.	camera / Leap motion	percussion stroke time	ens.
Metatone Classifier [50]	update interface during free improvisation	RF, Markov	control data	improvisation events	ens.
Neural Touch-screen Ensemble [51]	accompany improvisation in real-time	RNN, RF	control data	ensemble gesture classes	ens.
Orchestra in a box [63]	score-following system	HMM, Bayes'n net	audio	score locations	ens.
TablaNet [67]	tabla stroke recognition and phrase prediction	kNN, Bayes'n net	audio	symbolic music	ens.

Table 1: Predictive interactive music systems reviewed in this paper ordered by type: instrument (inst.), musical (music.), and ensemble (ens.). The machine-learning model used, and its input and output configuration are also listed.

Title	Description	Model	Input	Output	Type
Shimon [9]	robotic marimba player	Markov, curve matching,	kinect, symbolic music	physical movements	ens.
SHEILA [77]	robotic drum player	ESN	drum pattern class	symbolic music, motor control	ens.

References

[1] N. S. Altman. An introduction to kernel and nearest-neighbor non-parametric regression. *The American Statistician*, 46(3):175–185, 1992. `doi:10.1080/00031305.1992.10475879`.

[2] C. Ames. Automated composition in retrospect: 1956-1986. *Leonardo*, 20(2):169–185, 1987. `doi:10.2307/1578334`.

[3] C. Ames. The Markov process as a compositional model: A survey and tutorial. *Leonardo*, 22(2):175–187, 1989. `doi:10.2307/1575226`.

[4] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. Omax brothers: A dynamic topology of agents for improvization learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, pages 125–132, New York, NY, USA, 2006. ACM. `doi:10.1145/1178723.1178742`.

[5] G. Assayag and S. Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, 8(9):604–610, Sep 2004. `doi:10.1007/s00500-004-0385-4`.

[6] M. Ben-Asher and C. Leider. Toward an emotionally intelligent piano: Real-time emotion detection and performer feedback via kinesthetic sensing in piano performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 21–24, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST. URL: http://nime.org/proceedings/2013/nime2013_48.pdf.

[7] J. A. Biles. Improvizing with genetic algorithms: Genjam. In E. R. Miranda and J. A. Biles, editors, *Evolutionary Computer Music*, pages 137–169. Springer London, London, 2007. `doi:10.1007/978-1-84628-600-1_7`.

[8] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello. Deep salience representations for f_0 estimation in polyphonic music. In *18th International Society for Music Information Retrieval Conference*, pages 63–70, 2017.

[9] M. Bretan, M. Cicconet, R. Nikolaidis, and G. Weinberg. Developing and composing for a robotic musician using different modes of interaction. In *Proceedings of the International Computer Music Conference*, 2012. URL: http://hdl.handle.net/2027/spo.bbp2372.2012.092.

[10] M. Bretan and G. Weinberg. A survey of robotic musicianship. *Commun. ACM*, 59(5):100–109, Apr. 2016. URL: http://doi.acm.org/10.1145/2818994, `doi:10.1145/2818994`.

[11] A. R. Brown and T. Gifford. Prediction and proactivity in real-time interactive music systems. *Int. Workshop on Musical Metacreation*, pages 35–39, 2013. URL: http://eprints.qut.edu.au/64500/.

[12] J. Bullock and A. Momeni. ml.lib: Robust, cross-platform, open-source machine learning for max and pure data. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 265–270, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University. URL: http://www.nime.org/proceedings/2015/nime2015_201.pdf.

[13] J.-P. Cáceres, R. Hamilton, D. Iyer, C. Chafe, and G. Wang. To the edge with China: Explorations in network performance. In *ARTECH 2008: Proc. 4th Int. Conf. Digital Arts*, pages 61–66, 2008.

[14] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua. Adaptive gesture recognition with variation estimation for interactive systems. *ACM Transactions on Interactive Intelligent Systems*, 4(4):18:1–18:34, 2014. `doi:10.1145/2643204`.

[15] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6):759 – 771, 1991. `doi:https://doi.org/10.1016/0893-6080(91)90056-B`.

[16] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural networks*, 5(6):961–970, 1992.

[17] A. Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204, 2013. `doi:10.1017/S0140525X12000477`.

[18] B. A. Clegg, G. J. DiGirolamo, and S. W. Keele. Sequence learning. *Trends in Cognitive Sciences*, 2(8):275–281, 1998. URL: http://dx.doi.org/10.1016/S1364-6613(98)01202-9, `doi:10.1016/S1364-6613(98)01202-9`.

[19] D. Conklin. Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

[20] R. B. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *Commun. ACM*, 49(8):38–43, Aug. 2006. URL: http://doi.acm.org/10.1145/1145287.1145311, `doi:10.1145/1145287.1145311`.

[21] S. Davies. *Themes in the Philosophy of Music*. Oxford University Press, Oxford, UK, 2005.

[22] J. Drummond. Understanding interactive systems. *Organised Sound*, 14:124–133, 8 2009. doi:10.1017/S1355771809000235.

[23] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine-learning methods for musical style modeling. *Computer*, 36(10), October 2003. doi:10.1109/MC.2003.1236474.

[24] D. Eck and J. Schmidhuber. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In *Proc. 12th IEEE Workshop on Neural Networks for Signal Processing*, pages 747–756, 2002. doi:10.1109/NNSP.2002.1030094.

[25] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. *ArXiv e-prints*, Apr. 2017. URL: https://arxiv.org/abs/1704.01279.

[26] S. S. Fels and G. E. Hinton. Glove-talkii-a neural-network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Transactions on Neural Networks*, 9(1):205–212, Jan 1998. doi:10.1109/72.655042.

[27] R. Fiebrink. Machine learning as meta-instrument: Human-machine partnerships shaping expressive instrumental creation. In T. Bovermann, A. de Campo, H. Egermann, S.-I. Hardjowirogo, and S. Weinzierl, editors, *Musical Instruments in the 21st Century: Identities, Configurations, Practices*, pages 137–151. Springer Singapore, Singapore, 2017. doi:10.1007/978-981-10-2951-6_10.

[28] 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, 2011. ACM. doi:10.1145/1978942.1978965.

[29] 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*, NIME '09, pages 280–285, 2009. URL: http://www.nime.org/proceedings/2009/nime2009_280.pdf.

[30] N. Gillian, R. B. Knapp, and S. O'Modhrain. A machine learning toolbox for musician computer interaction. In A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '11, pages

343–348, Oslo, Norway, 2011. University of Oslo. URL: http://www.nime.org/proceedings/2011/nime2011_343.pdf.

[31] A. Graves. Generating Sequences With Recurrent Neural Networks. *ArXiv e-prints*, Aug. 2013. URL: https://arxiv.org/abs/1308.0850, `arXiv:1308.0850`.

[32] D. Ha and D. Eck. A Neural Representation of Sketch Drawings. *ArXiv e-prints*, Apr. 2017. URL: https://arxiv.org/abs/1704.03477.

[33] G. Hadjeres, F. Pachet, and F. Nielsen. DeepBach: a steerable model for Bach chorales generation. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1362–1371, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL: http://proceedings.mlr.press/v70/hadjeres17a.html.

[34] P. Hamel and D. Eck. Learning features from music audio with deep belief networks. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, ISMIR '10, pages 339–344, 2010.

[35] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag New York, New York, USA, 2009. `doi:10.1007/978-0-387-84858-7`.

[36] J. Hawkins and S. Blakeslee. *On Intelligence*. Times Books, New York, NY, USA, 2004.

[37] A. Hawryshkewich, P. Pasquier, and A. Eigenfeldt. Beatback: A real-time interactive percussion system for rhythmic practise and exploration. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 100–105, Sydney, Australia, 2010. URL: http://www.nime.org/proceedings/2010/nime2010_100.pdf.

[38] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. `doi:10.1162/neco.1997.9.8.1735`.

[39] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003. `doi:10.1076/jnmr.32.4.429.18853`.

[40] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*,

volume 70 of *Proceedings of Machine Learning Research*, pages 1645–1654, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL: http://proceedings.mlr.press/v70/jaques17a.html.

[41] Z. Jin, R. Oda, A. Finkelstein, and R. Fiebrink. Mallo: A distributed synchronized musical instrument designed for internet performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '15, pages 293–298, 2015. URL: http://www.nime.org/proceedings/2015/nime2015_223.pdf.

[42] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. Blog post, May 2015. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[43] J. Lazzaro and J. Wawrzynek. A case for network musical performance. In *Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '01, pages 157–166, New York, NY, USA, 2001. ACM. doi:10.1145/378344.378367.

[44] M. Lee, A. Freed, and D. Wessel. Real-time neural network processing of gestural and acoustic signals. In *Proceedings of the International Computer Music Conference*. International Computer Music Association, 1991.

[45] B. Levy, G. Bloch, and G. Assayag. Omaxist dialectics: Capturing, visualizing and expanding improvisations. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan. URL: http://www.nime.org/proceedings/2012/nime2012_87.pdf.

[46] H. Lim, S. Rhyu, and K. Lee. Chord generation from symbolic melody using BLSTM networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017. URL: https://arxiv.org/abs/1712.01011.

[47] I. Malik and C. H. Ek. Neural translation of musical style. *ArXiv e-prints*, Aug. 2017. URL: https://arxiv.org/abs/1708.03535.

[48] Y. Mann. Ai duet. Interactive Web Page, 2016. URL: https://aiexperiments.withgoogle.com/ai-duet.

[49] M. Marchini, F. Pachet, and B. Carré. Rethinking reflexive looper for structured pop music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 139–144, Copenhagen, Denmark, 2017. Aalborg University Copenhagen.

[50] C. Martin, H. Gardner, and B. Swift. Tracking ensemble performance on touch-screens with gesture classification and transition matrices. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '15, pages 359–364, Baton Rouge, LA, USA, 2015. Louisiana State University. URL: http://www.nime.org/proceedings/2015/nime2015_242.pdf.

[51] C. P. Martin, K. O. Ellefsen, and J. Torresen. Deep models for ensemble touch-screen improvisation. In *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*, AM '17, pages 4:1–4:4, New York, NY, USA, 2017. ACM. `doi:10.1145/3123514.3123556`.

[52] C. P. Martin and J. Torresen. Exploring social mobile music with tiny touch-screen performances. In T. Lokki, J. Pätynen, and V. Välimäki, editors, *Proceedings of the 14th Sound and Music Computing Conference*, SMC '17, pages 175–180, Espoo, Finland, 2017. Aalto University. URL: http://smc2017.aalto.fi/media/materials/proceedings/SMC17_p175.pdf.

[53] C. P. Martin and J. Torresen. RoboJam: A musical mixture density network for collaborative touchscreen interaction. In A. Liapis, J. J. Romero Cardalda, and A. Ekárt, editors, *Computational Intelligence in Music, Sound, Art and Design: International Conference, EvoMUSART*, volume 10783 of *Lecture Notes in Computer Science*, Switzerland, Apr. 2018. Springer International Publishing. URL: http://arxiv.org/abs/1711.10746, `arXiv:1711.10746`, `doi:10.1007/978-3-319-77583-8_11`.

[54] K. A. McMillen. Stage-worthy sensor bows for stringed instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 347–348, Genoa, Italy, 2008. URL: http://www.nime.org/proceedings/2008/nime2008_347.pdf.

[55] N. Mor, L. Wolf, A. Polyak, and Y. Taigman. A Universal Music Translation Network. *ArXiv e-prints*, May 2018. URL: https://arxiv.org/abs/1805.07848.

[56] M. C. Mozer. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. *Connection Science*, 6(2-3):247–280, 1994. `doi:10.1080/09540099408915726`.

[57] R. Oda, A. Finkelstein, and R. Fiebrink. Towards note-level prediction for networked music performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '13, pages

94–97, 2013. URL: http://nime.org/proceedings/2013/nime2013_258.pdf.

[58] N. Orio, S. Lemouton, and D. Schwarz. Score following: state of the art and new developments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '03, pages 36–41, Montreal, Canada, 2003. McGill University. URL: http://www.nime.org/proceedings/2003/nime2003_036.pdf.

[59] F. Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003. doi:10.1076/jnmr.32.3.333.16861.

[60] F. Pachet, P. Roy, J. Moreira, and M. d'Inverno. Reflexive loopers for solo musical improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2205–2208, New York, NY, USA, 2013. ACM. doi:10.1145/2470654.2481303.

[61] K. A. Pati, S. Gururani, and A. Lerch. Assessment of student music performances using deep neural networks. *Applied Sciences*, 8(4), 2018. doi:10.3390/app8040507.

[62] L. S. Petroa and L. Mucklia. The brain's predictive prowess revealed in primary visual cortex. *Proceedings of the National Academy of Sciences*, 113(5), 2016. doi:10.1073/pnas.1523834113.

[63] C. Raphael. Orchestra in a box: A system for real-time musical accompaniment. In *Proc. Int. Joint Conf. on AI (Working Notes of IJCAI-03 Rencon Workshop)*, pages 5–10, 2003. URL: http://music.informatics.indiana.edu/~craphael/papers/ijcai03.pdf.

[64] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*, 2018.

[65] S. Ross and N. C. Hansen. Dissociating prediction failure: Considerations from music perception. *Journal of Neuroscience*, 36(11):3103–3105, 2016. doi:10.1523/JNEUROSCI.0053-16.2016.

[66] R. Rowe. *Interactive Music Systems: Machine Listening and Composing.* The MIT Press, 1993. URL: https://wp.nyu.edu/robert_rowe/text/interactive-music-systems-1993/.

[67] M. Sarkar and B. Vercoe. Recognition and prediction in a network music performance system for indian percussion. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '07, pages 317–320, 2007. doi:10.1145/1279740.1279809.

[68] M. Schedel, P. Perry, and R. Fiebrink. Wekinating 000000swan: Using machine learning to create and control complex artistic systems. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '11, pages 453–456, 2011. URL: http://www.nime.org/proceedings/2011/nime2011_453.pdf.

[69] I. Simon, D. Morris, and S. Basu. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 725–734, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1357054.1357169, doi:10.1145/1357054.1357169.

[70] I. Simon and S. Oore. Performance rnn: Generating music with expressive timing and dynamics. https://magenta.tensorflow.org/performance-rnn, 2017.

[71] B. D. Smith and G. E. Garnett. The self-supervising machine. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 108–111, Oslo, Norway, 2011. URL: http://www.nime.org/proceedings/2011/nime2011_108.pdf.

[72] J. Snyder. The birl: Adventures in the development of an electronic wind instrument. In T. Bovermann, A. de Campo, H. Egermann, S.-I. Hardjowirogo, and S. Weinzierl, editors, *Musical Instruments in the 21st Century: Identities, Configurations, Practices*, pages 181–205. Springer Singapore, Singapore, 2017. doi:10.1007/978-981-10-2951-6_13.

[73] J. Snyder and D. Ryan. The birl: An electronic wind instrument based on an artificial neural network parameter mapping structure. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 585–588, London, United Kingdom, 2014. Goldsmiths, University of London. URL: http://www.nime.org/proceedings/2014/nime2014_540.pdf.

[74] B. L. Sturm and O. Ben-Tal. Taking the models back to music practice: Evaluating generative transcription models built using deep learning. *Journal of Creative Music Systems*, 2(1), 2017. URL: http://jcms.org.uk/issues/Vol2Issue1/taking-models-back-to-music-practice/article.html.

[75] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova. Music transcription modelling and composition using deep learning. In *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*, 2016. URL: http://arxiv.org/abs/1604.08723.

[76] R. Sun and C. L. Giles. Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*,

16(4):67–70, 2001. URL: http://dx.doi.org/10.1109/MIS.2001.1463065, doi:10.1109/MIS.2001.1463065.

[77] A. Tidemann, P. Öztürk, and Y. Demiris. A groovy virtual drumming agent. In Z. Ruttkay, M. Kipp, A. Nijholt, and H. H. Vilhjálmsson, editors, *Intelligent Virtual Agents, IVA 2009*, volume 5773 of *Lecture Notes in Computer Science*, pages 104–117. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-04380-2_14.

[78] A. Van, B. Caramiaux, and A. Tanaka. PiaF: A tool for augmented piano performance using gesture variation following. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '14, pages 167–170, 2014. URL: http://www.nime.org/proceedings/2014/nime2014_511.pdf.

[79] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *ArXiv e-prints*, abs/1609.03499, Sept. 2016. URL: http://arxiv.org/abs/1609.03499, arXiv:1609.03499.

[80] B. Yuksel, D. Afergan, E. Peck, G. Griffin, L. Harrison, N. Chen, R. Chang, and R. Jacob. BRAAHMS: A novel adaptive musical interface based on users' cognitive state. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '15, pages 136–139, 2015. URL: http://www.nime.org/proceedings/2015/nime2015_243.pdf.