arXiv:1801.10492v1 [cs.SD] 31 Jan 2018

Deep Predictive Models in Interactive Music

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

Abstract

Automatic music generation is a compelling task where much recent progress has been made with deep learning models. In this paper, we ask how these models can be integrated into interactive music systems; how can they encourage or enhance the music making of human users? Musical performance requires prediction to operate instruments, and perform in groups. We argue that predictive models could help interactive systems to understand their temporal context, and ensemble behaviour. Deep learning can allow data-driven 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 incorporating such predictive models into the sensing, processing, and result architecture that is often used in musical interface design. We show that our framework accommodates deep generative models, as well as models for predicting gestural states, or other high-level musical information. We motivate the framework with two examples from our recent work, as well as systems from the literature, and suggest 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, 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 predicting unknown or future states of the world from our current and past predictions. When our predictions are not accurate, such as lifting our feet for one too many steps, the error is used

*Department of Informatics, RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion, University of Oslo, 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

as a warning to correct our actions, in that case, the sensation of a surprise. Neuroscientists are now able to observe prediction in action in the human brain. In particular, prediction has been observed for visual perception [36], as well as musical perception [38]. Other researchers have theorised that prediction and expectations are key to our aesthetic appreciations [6], and, indeed, that prediction is the fundamental basis for intelligence [21].

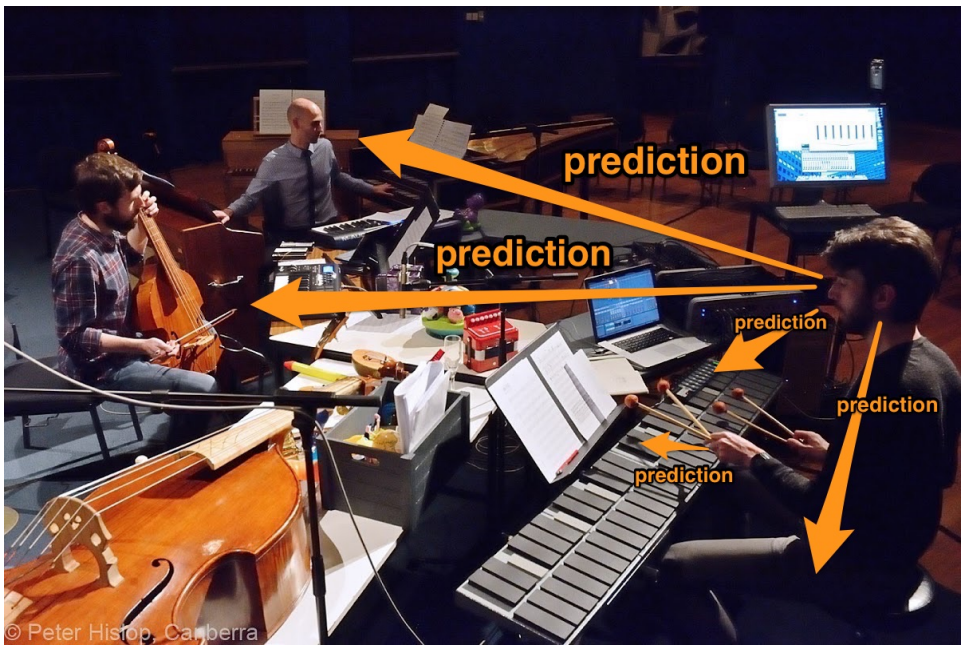

Figure 1: Many layers of prediction are required in musical performance from low-level instrumental control to high-level planning using multiple senses. The predictions of musical deep learning models can be used in digital instruments to support performers. (Photo: Peter Hislop)

Naturally, musical performance involves many layers of prediction (see Figure 1): Skilled performers predict the sounds produced by different instrumental gestures, predict the musical effect of rehearsed expressions and improvised sounds, predict the musical actions of an ensemble, and predict the response of an audience. It may seem natural that interactive music systems should incorporate prediction in order to better account for the complexity of musical performance; however, as Brown and Gifford have noted, prediction has only been modestly implemented in such systems [6]. These authors focused on the potential for prediction to be used to incorporate proactivity in musical agents, in contrast, we feel that predictive models may have many applications in interactive music systems that can improve their integration into creative environments and provide new possibilities for musical interaction. In fact, some DMIs and NIMEs already use predictive

models of various kinds, particularly when they seek to generate new musical data, manage ensemble experiences, or handle complex sensor input. The design frameworks that are often called upon to understand these interactive music systems do not generally consider the role of prediction; they tend to focus on *reactive* rather than *predictive* operation.

In this paper we argue that interactive music designs using predictive models can lead to significant new affordances for users. Some concepts from artificial intelligence (AI), such as generative music [2] or machine learning (ML) [9] can used as predictive models in interactive music systems; however, we argue that recent work in deep learning, particularly generative sequence models, can have particularly meaningful applications in musical interaction and performance, and not just in music composition.

As we will discuss later in this paper, predictive interactive music designs are able to fill in unknown data for performers and users, rather than directly interpret control from a user interface. There are many potential benefits to predictive interfaces: They could fill in, or improve, aspects of music such as melody, harmony, or rhythm that the user is unable to control. They could be proactive, triggering events before the user explicitly asks for them. Complex user expressions, such as body motions captured by multiple sensors, can be incorporated into a musical performance, or even used as fundamental controls. Finally predictive interfaces have the capacity to understand that music occurs on a temporal axis. Future musical events have significant dependencies on past actions; but, as expert human musicians show us, they are quite predictable. Incorporating predictive models into DMIs and NIMEs would allow us to take advantage of much useful musical data.

In this research we argue that deep learning models have much to offer interactive music design. Recurrent neural networks (RNNs) have the capacity to learn a large amount of general musical information, but still be conditioned to perform in a specific musical context, such a user's recent playing style. Such models are also highly flexible in terms of their architecture and can be trained to model a number of inputs simultaneously, or multiple dimensions of real-valued data. This ability is useful when predicting an ensemble situation with multiple performers, or data from multiple sensors. Overall, present work in deep learning emphasises an entirely data-driven approach to ML in contrast with previous approaches in generative music that often start from the rules of music theory. In a digital music context, gestural control data and synthesiser parameters may not be well modelled by music theory and so an entirely data-driven approach may be more appropriate.

Ensemble experiences are a valuable part of music making; however, support for this experience is not often build into interactive music systems. By participating in a group, musicians collaborate and experience the emergence of new musical ideas and interpretations [41] that cannot be reached in a solo performance. In this research we discuss two of our own systems where deep learning models are applied to simulate an ensemble experience,

either for practice, or for when other musicians are not available. These predictive models could have significant impact on users' experience with these interfaces and would not be possible without the deep models that were used.

In this research we introduce a general framework for how prediction can be implemented in a typical interactive music architecture, a concept that we call "predictive musical interaction". While our framework is developed around an abstract notion of prediction, we discuss where recent advances in deep learning can be applied. We discuss the application of our framework to two of our own interactive music systems, RoboJam and the Neural Touchscreen Ensemble, that apply RNNs to musical interaction. We also examine several systems from the literature that use deep learning as well as other ML models. The framework, and these systems, suggests future opportunities for endowing NIMEs with predictive intelligence and the ability to support more expressive, enjoyable, and democratic interactions.

In the next section we will develop a precise definition for predictive musical interaction and discuss 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 NIMEs. 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 Section 5, we will outline the benefits that predictive models can offer to NIME designers and users.

2 Models for Prediction and Interaction

As previously discussed, cognition involves many levels of prediction that we rely on for our everyday actions; however, it is not always clear how prediction could be integrated into creative tools in a beneficial way. In order to develop a framework for including predictive models in musical interaction, we wish to motivate how predictive models and interactive systems can work together, what is required in order for prediction to take place, and what musical predictive models show most promise for interactive use.

2.1 Interacting with Predictions

In this paper, we focus on a simple definition for prediction, that is, the estimation of unknown data based on current knowledge and perceptions. How, then, can prediction be used in an interactive system? What more information can be required than commands given by a user? We can motivate our definition of predictive interaction and future examples by presenting the forward model, a simple bio-inspired predictive system that is often used in real-time control of robots.

Humans and animals rely on internal models to simulate themselves and the environment, allowing them to predict the consequences of their actions and the behaviour of others [52, 53]. Inspired by the human brain, robotic and computer systems have been designed with similar models – enabling cognitive skills such as prediction, behaviour recognition and imitation [43]. The forward model predicts the sensory outcomes of an action, using an activity-generating signal (e.g., a motor command) as input [53]. Such prediction is thought to take place in the cerebellum in human brains [5].

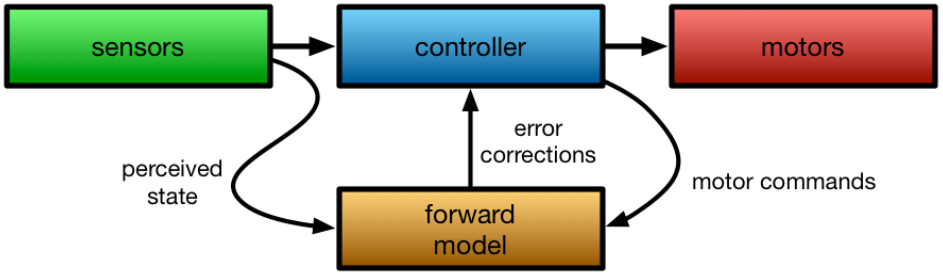

Figure 2: A forward model applied in a robot control architecture. The forward model predicts future sensory input from motor commands. Errors between the predictions and reality can be used to correct the control system and react to unexpected circumstances.

Figure 2 demonstrates how a forward model could be integrated in a robot control architecture. The effects of motor commands are predicted by the forward model, and simultaneously carried out by the motor system. Discrepancies between the predicted and actual outcome can be used to correct the controller's behaviour. In our case, this example serves to show how a predictive model can be embedded into a real-time system to check whether mappings from sensors to actuators make sense. In the context of an interactive music system that maps sensor inputs to musical notes, a forward model could be used to predict whether outputs make *musical* sense. For instance, the predictive model could generate appropriate subsequent notes, based on the those previously played and the rules of music theory, and update the controller to make such notes easier to play.

2.2 What is a Prediction?

Forward models predict sensory input in the next time step, which could be termed S_{t+1}, from observations at the current time step, S_t, and the current motor command, M_t [43]. In this case, the predictions are in the same space as existing data, gathered from sensors, but are in a different and unknown temporal location, the future. In a musical application, the data sources for prediction may vary, but a similar model of temporal prediction can also

be defined. In Figure 3, we define the predictive model as mapping some current state, S_t, along with some additional information, I_t, to a predicted future state S_{t+1}. Depending on the specifics of the prediction problem, that information may include a model of a user (e.g., their playing style), a history of the users' previous actions or preferences, or information about the musical context. In ML, this kind of temporal prediction is often referred to as sequence learning [47] or time series forecasting [10].

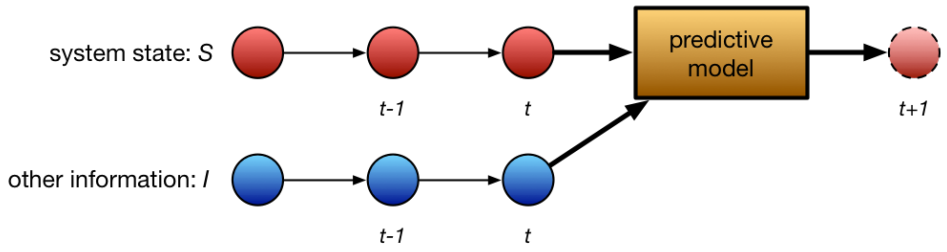

Figure 3: A temporal predictive model infers the future state of a system (S_{t+1}), from the current state (S_t), and other available information (I_t)

We should be careful to note that predictions are not always temporal. In ML, the two typical types of prediction tasks are classification and regression, where models are trained to predict categorical and quantitative data respectively [20]. Both of these tasks are most often applied when predicting a different type of data than that given as input, without supposing any temporal relationship. Indeed, non-temporal predictions can have a role in musical interaction as well, for instance, a model might predict harmony from a present melody. An example for non-temporal prediction is given in Figure 4. In this example, the system state S_t, is used to predict some other unknown information U_t. In this case, past values of S and previous predictions of U are not used to predict the present U_t; however, a temporal model could make use this history to make better predictions for U.

Indeed, while temporal predictions would seem to be most relevant in a temporal artform such as music, most existing NIMEs and DMIs that do use predictive interaction do so in a non-temporal way. Thus, the recent progress in deep models that can learn long-range temporal relationships could have a significant impact on the possibilities for predictive musical interaction.

It is worth considering here what general conditions should be met in an interactive music system to make use of a predictive model. A precondition for generating predictions is *perception of the environment*, allowing information about S_t and I_t to be collected. We are here using the word environment in its most general form, meaning anything we would like to make predictions about, for instance, users' inputs to a musical instrument.

Further, temporal prediction requires a *temporal model*, which can hold the information about how the environment changes from one time-step to the

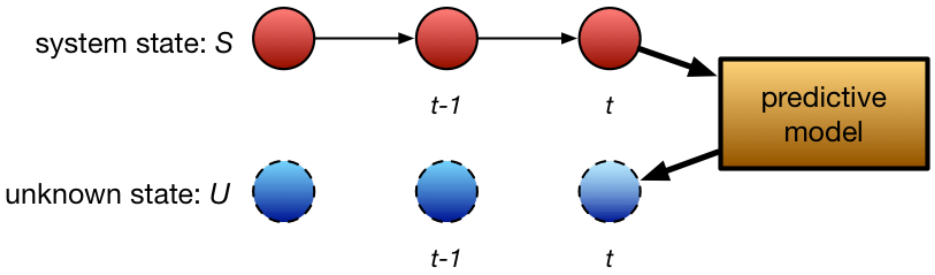

Figure 4: Predictive models can also infer unknown information (U_t) from the known system state (S_t). In many cases, this prediction is made without referring to any temporal relationship in the sequence

next. In simple cases, this model may be pre-specified, but to make accurate predictions in the real world, the temporal model may need to be *learned* by observing several examples of the predicted phenomenon. Although it is not necessary for prediction, errors in the model's predictions may also be corrected in real-time.

Finally, an interactive music system must have some flexibility or freedom in its behaviour in order to actually use the result of predictions. Common NIME design advice favours one-to-one mappings between gesture, or sensor input, and result [37] in order to make sure that musical interfaces behave clearly and predictably. An alternative design idiom that embraces the potential for predictive interaction, would be to include an *action-space* of multiple mappings, or multiple potential outputs. An individual from this space could be selected according to the output of a predictive model. Each member of this space would constitute a musically sensible mapping; however, switching between them could be governed by continuous predictions about which would be most musically appropriate or stimulating.

2.3 Neural 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. RNNs have often been 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 [33] is an early example of this idea. The later introduction of gated units such as the long short-term memory cell [22] 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 [13]. These recurrent models have a flexible ability to learn about the temporal context in a sequence and thus mimic in part human cognitive abilities for sequence learning and prediction [11].

Other popular systems for generating music use Markov models to generate the emission probabilities of future notes based on those preceding [3]. The advantage of RNN models over Markov systems is the latter require unreasonably large transition tables in order to learn distant dependencies in the data, a point made by Mozer [33]. Graves subsequently commented that RNNs can make more "fuzzy" predictions than Markov systems given that predictive distributions are not simply given by matching the recent past to a transition table [17].

The proliferation of GPU computation and large datasets in recent times have contributed to the popularity of creative RNN models. Character-level text generation, often inspired by Karpathy's description of a CharRNN [25], is now well known in computational arts. Music, too, can be represented as text and represented in similar RNN designs as were the "ABC" formatted folk songs of Sturm et al.'s FolkRNN project [46]. 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 allow such a model to be steered towards generating voices to accompany certain melodies [19]. RNN models can even be combined with the rules of music theory via a reinforcement learning tuning step described by Jaques et al. [23]. Google's Magenta project[1] have developed a collection of RNN models for music generation and have notably released trained versions of several musical RNN models as well as tools for accessing these models through popular music production software.

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. [46] acknowledge as much, calling the output of FolkRNN "transcriptions" of (potential) folk tunes, not tunes themselves. These transcriptions have a melody, but to perform them, musicians need to contribute their own arrangement and expression in order to effectively communicate this information. 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 [26] annotates existing musical scores with dynamic (volume) markings. Simon and Oore's PerformanceRNN [45] 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* [12] than FolkRNN's

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

thin output because it contains much more information required for to ultimately perform the a musical work.

Of course, an even thicker representation of music would be the actual sounds of the performance. ANN models that directly predict digital audio signals could generate music at this level. The WaveNet model [51] can render raw audio samples using dilated causal convolutional layers, rather than a recurrent network, to handle temporal dependencies. While WaveNet has been shown to be able to produce musical sounds, computational requirements have not been sufficiently overcome for this to be widely explored. A more manageable task was demonstrated with Nsynth [14], a WaveNet model applied to generating musical samples, short recordings of single instrumental notes that can used to synthesise whole performances. Nsynth applied ANN generation to timbre, or tone colour, and can be used to explore the latent space in between instrumental sounds. While Nsynth doesn't produce complete musical works, it can potentially be used with other musical models in a workflow analogous to the process of composing and performing a new piece of music.

The above ANN models of music and audio have progressed from relatively simple representations of melody and harmony to include expression and even generate complete performance recordings. In the rest of the paper, we will discuss how ML and ANN models have and can be integrated into interactive music systems. These systems will apply these models to predict a variety of training data, including MIDI notes as well as gestural control data, in solo and ensemble performance scenarios.

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 predictive models as an important part of interaction. This framework divides interactive music systems into three stages: sensing, processing, and response, as shown in Figure 5, and was originally due to Rowe [39]. While this framework is simple, it provides a helpful division of concerns and has often been used to frame NIME and DMI designs. 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 as is the case with the many MIDI controller and synthesiser devices available.

In the context of research-focussed musical interfaces, the elements of this system are often prototype electronic systems, and bespoke software using computer music environments. 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 a Raspberry Pi in the processing stage, and

the response could be audio send to a speaker system. Such an arrangement is illustrated in Figure 5.

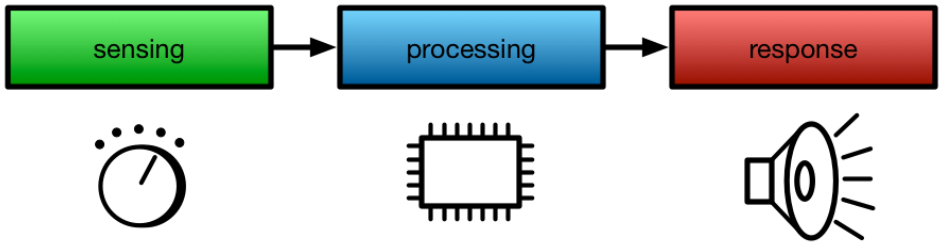

Figure 5: Rowe's three-stage model for interactive music systems.

Of course, the way that components of an interactive music system are assigned to this simple framework may be somewhat flexible. In particular, many parts of a typical NIME could be considered part of "processing" but it may be more practical to restrict this stage to high level mappings from sensors to musical parameters. For instance, raw sensor data might be interpreted by a microcontroller before being sent to a synthesis environment. Similarly, most music systems output digital audio at the response, but we often consider a high level musical signal such as MIDI as a response, even though this would require further processing by a synthesiser to generate sound.

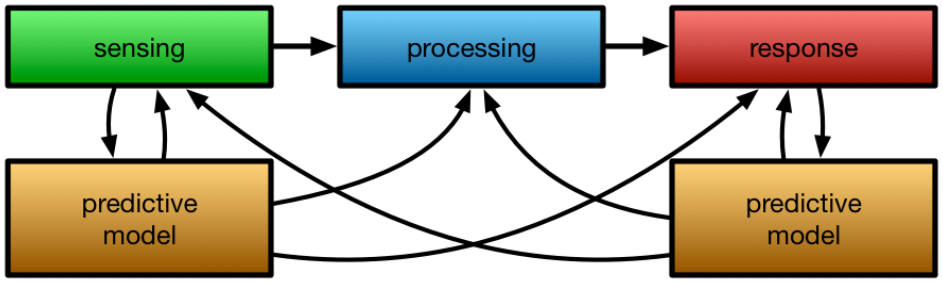

Figure 6: The interactive music system expanded with possible locations for predictive models.

In Figure 6, we consider where a predictive model could function in this framework. Clearly the model could take the sensor data as input, in this position it would be predicting the actions of the performer or sensed environment. A predictive model could also be placed at the response stage where it would predict the high-level musical output of the system. The output from these predictive models, either future states of the input data, or the present value of some other unknown part of the system, can then be used at either the sensing, processing, or response stages.

Figure 7: A predictive model can replace the processing stage in the framework. This type of prediction usually would be of the non-temporal type illustrated in Figure 4.

A predictive model can also simply replace the processing stage. This processing stage can be considered a mapping between input and output data. Given that the output data is unknown, a non-temporal predictive model, could serve as the processing stage. This scenario is illustrated in Figure 7. In fact, as described below, this is one of the most common uses for predictive models in interactive music design.

There are many options as to how the outputs of a predictive model could be used at the three stages of the system, but examples working from right to left could be as follows:

- Predicted data sent to the response stage could result in a system that "plays itself" in some way, or accompanies the performer;
- Feedback to the processing stage could change how the user's input is interpreted, perhaps by selecting different synthesiser parameters;
- Feedback to the sensing stage could change the morphology of the instrument itself: in a virtual reality or touchscreen instrument, the controls visible to the performer could be updated, or in a haptic interface, the feedback given to the performer could change.

Recalling from Figure 1 that interacting with other musicians in an ensemble requires many different predictions, we can expand the framework to include an ensemble experience. Figure 8 shows how predictive models could help to connect multiple interactive music systems. The predictive models take input from one player's system and could send predictions to any part of the other systems. The predictions here could be used in a similar way to the solo framework above; however, where they were used above to enhance or extend one user's creativity, they could be used here to help an ensemble collaborate, understand each other more clearly, or even to generate an artificial ensemble experience for a solo performer.

4 Predictive Interactive Music Designs

In this section, we will discuss how predictive models can be incorporated into interactive music designs following the framework given in Section 3. To motivate our discussion we refer to two examples of work from our

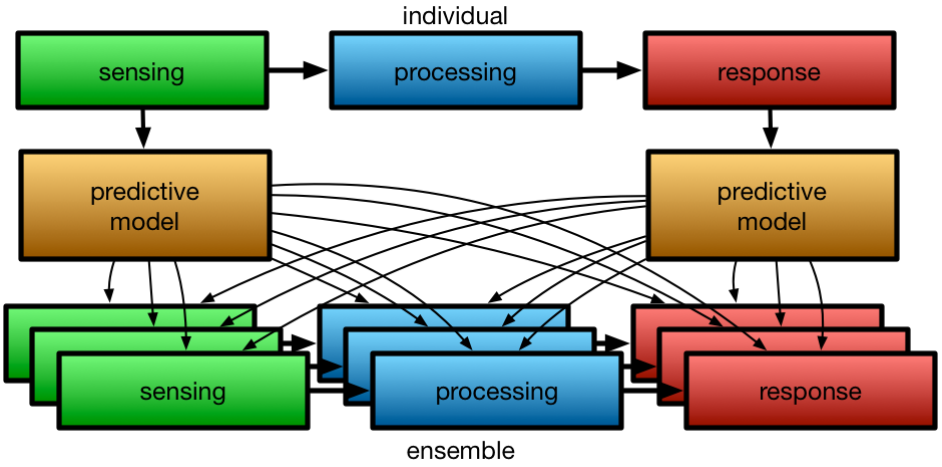

Figure 8: Predictive models could use one performer's system to influence other ensemble members' musical interactions.

group as well as examples of existing interactive music systems that use predictive models. 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 Predicting the input

The first potential location for a predictive model is at the sensing stage of an interactive music system. In this case, the input from the predictive model is data from the sensors or interface used to interact with the music system, that is, the same data as is sent to the processing stage. An advantage of placing predictive models at the sensing stage is that they are as close to detecting the user's actual control movements as possible. A model here can make embodied predictions about the user's performance intentions that may not be initially clear in the data. It could also generate more control data than provided by the user in order to continue or respond to performances.

4.1.1 PiaF (Piano Follower)

The PiaF or Piano Follower [50] 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 (sensing) connected to an audio processing system with sound output as the response. 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 [8]. This temporal predictive model tracks multiple dimensions of input data in order to classify from a number of trained gestures. In addition to predicting which discrete gesture is being performed, it provides continuous data about the speed, scale and rotation of the gestural data. 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 predictive model thus takes sensing data and uses predictions to update parameters in the processing stage. The result is a system that can enhance the pianist's expressive options during performance.

4.1.2 MalLo

The MalLo system aims to support low-latency percussion performances over long-distance network connections by incorporating a predictive model into a percussion instrument [24]. This model, described by Oda et al. [34], uses computer vision techniques to track the position of the percussion performer's mallets. The model is able to predict when the mallet will strike the instrument before this actually occurs. In local performances, typical reactive electronic percussion instruments (e.g., digital drum sets) have a comfortably low gesture-to-sound latency. However, in networked performances over a long distance, the latency is often too high to support a fast tempo. 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 is an example of a predictive system that can support ensemble performances. The MalLo predictive model senses the remote performer's mallet position, and outputs note signals at the local performer's response stage. The temporal model is MalLo's model of mallet position in time which predicts the likelihood and timing of an upcoming strike. Networked ensemble performance is one area where interactive music designs can clearly benefit from temporal predictive models. Similar systems have been implemented to predict Indian percussion patterns [40], and to support massed ensemble performances using a common metronome [7].

4.1.3 MySong

MySong is a system to automatically generate harmony accompaniments for vocal melodies [44]. 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 complete 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.

Here, prediction occurs at the input stage, as the model takes the user's direct audio input and annotates it with unknown information, the chords. The output of the predictive model is also control data which can then be processed into accompaniment tracks and played back simultaneously with the vocal performance. 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.

4.1.4 RoboJam

RoboJam [31] is a call-and-response agent for predicting responses to touch-screen music performed in a social smartphone app [30]. To accomplish this, RoboJam uses an RNN conditioned on a human performance to generate a response. While many systems use an RNN to model musical notes, RoboJam is unique in using an RNN to model musical control data. 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. In general, the x-position of the touch determines pitch of a sound, and the y-position determines other parameters of expression. Once recorded and uploaded the performances can be played back by other users along with an animation of the user's touch patterns. To collaborate, users can "reply" to performances–by recording a response–which is layered over the top of the original such that both performances play simultaneously. In this way, users can create complex, if short, compositions with simple creative mappings. A complication with this system is that the collaborative experience relies on interaction with another user. This presents an opportunity for a predictive model to generate potential responses quickly. RoboJam provides this

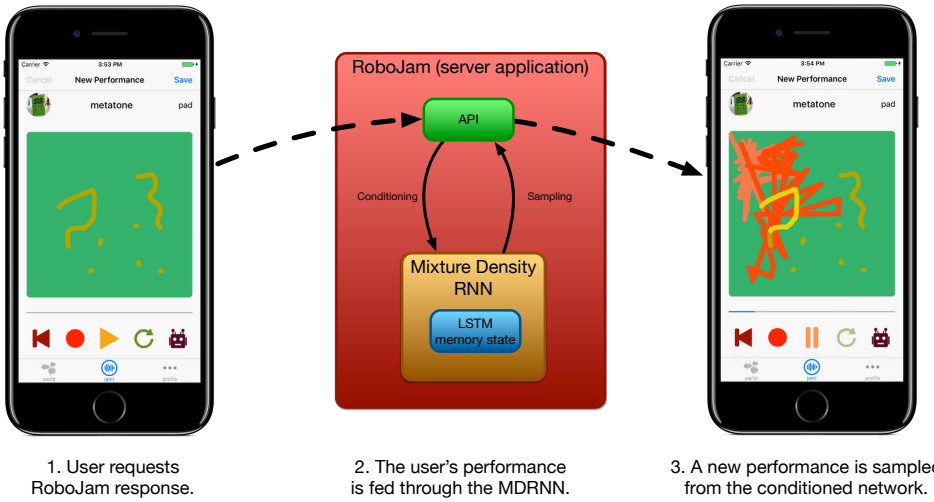

Figure 9: RoboJam is a call-and-response agent for predicting responses to touchscreen musical performances. It uses a mixture density RNN to generate a sequence of real-valued touch interactions after being conditioned on a user's performance.

functionality by using an RNN to automatically generate responses from a user's performance, these are then played back in layers with RoboJam's performances played back with a different synthesised sound from the original. These responses can be generated whenever, and however many times they are required. The resulting interaction allows users to hear more complex performances quickly, and also, by generating multiple responses, to hear their performance in context with different layered sounds.

RoboJam's predictive model is trained on a large corpus of 4.3M musical touchscreen interactions, each interaction consisting of a triple, (x, y, dt) representing the touch location on the touchscreen, and a time delta since the previous interaction. The model is set up to predict these interactions one at a time in order to generate extended sequences. To accomplish this, the model uses of a typical RNN architecture with three layers of 512 LSTM units. This is followed by two separate mixture density networks (MDN). The first of these uses a mixture of 2D normal distributions to predict locations on the touchscreen, and the second uses a mixture of 1D normal distributions to predict the time delta until this interaction should be played. This ANN design is inspired by MDN models such as SketchRNN [18], which is able to create line drawings.

This RNN configuration has several important implications for the predictions related to the ability of the mixture models to predict real-valued outputs, rather than discrete classes. First, the temporal location of interactions can be predicted absolutely, rather than on a rhythmic grid. Much

music, particularly in the electronic dance music (EDM) genre, limits rhythmic placement of notes to 16 subdivisions of each measure. Musical RNN models often follow this formalism by predicting the presence or absence of notes in each subdivision. By predicting absolute time, RoboJam's MDN overcomes this limitation to support the "free" rhythms more typical of improvised touchscreen performance. RoboJam can also predict note values with higher precision than 128 standard MIDI pitches as its bi-dimensional mixture model predicts real values.

RoboJam predicts data at the sensing stage of the smartphone music app. Since it predicts musical control data, rather than notes, it could be said to learn how to *perform* music, than how to *compose*. The predictions are then used in the sensing stage of a parallel system, representing the agent responding to the user's performance.
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.

4.2 Prediction as processing

Replacing the processing stage of an interactive music system with a predictive model is one of the most widely explored uses of ML models in musical interaction. When an ML model is used as the processing stage, connections from sensing to response stages can be made by example, rather than hand coded, leading to several advantages. In designs with complex sensing arrangements, for example, a point cloud from a 3D camera, the predictive model might be able to learn to extract salient features from this information which can then be connected by the designer to synthesis or composition parameters. It should be noted that, in general, the models used in prediction as processing are not temporal.

4.2.1 Wekinator

Many artists and researchers wish to connect complex or multiple sensors to the parameter controls of audio or computer graphics systems. This can often be accomplished effectively with classical ML models such as k-nearest neighbour classification [1], or shallow ANNs. Wekinator [16] is a framework embedding such algorithms into interactive music systems and allowing them to be trained interactively and on-the-fly. In practice, training such models on-the-fly allows for valuable creative exploration of their affordances and predictive power [15].

A typical use case of Wekinator is seen in Schedel et al.'s performances

[42] where the output from a Kinect camera and a K-Bow, a sensor-laden bow for string instruments [32], are tracked by Wekinator's predictive models. 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. In performance, the predictive models take input at the sensing stage and sends output directly to the response stage completing the interactive system.

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. Similar frameworks could focus more on temporal predictive models to better account for temporal effects in performance, while maintaining the concept of on-the-fly training and evaluation.

4.2.2 SHEILA

The SHEILA (Software for Hierarchical Extraction and Imitation of Drum Patterns in a Learning Agent) system uses a predictive ML model to control an animated drummer. SHEILA [48] used ideas from biology and artificial intelligence to create a virtual drumming agent that imitates a human style in motion as well as rhythm. The system is trained on MIDI and motion capture recordings from a human drummer performing in a particular style and aims to imitate the characteristic "groove" of this performer.

SHEILA actually contains two models, one that generates MIDI drum notes, and another that controls the movements of the animated drumming agent. This second system maps notes to motor commands in the agent and uses a predictive forward model to predicts the limb position and correct errors [49]. We can consider SHEILA's sensing stage to be the generation of drum patterns, the processing stage to be the predictive mapping to motor commands that result in the animation and sound output.

4.3 Predicting responses

When a predictive model is placed at the response section of an interactive music system, it is able to make predictions from high-level musical data. While digital audio signals could be the ultimate response of most DMIs and NIMEs, here we consider the response to be high level outputs such as synthesiser commands, MIDI data, or gestural classes that are the output of some previous processing stage. By predicting higher level musical information, a predictive model can potentially learn abstract musical concepts

more easily, or understand the high-level musical structure of a performance.

4.3.1 The Continuator

The *Continuator* is a DMI that models and imitates the style of individual performers in order to "continue" their performances where they leave off [35]. 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 MIDI controller and Continuator software can be considered to be a single interactive music system with the output of MIDI notes as the response. In this case, the predictive model connects both its input and output to the response stage. 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.

4.3.2 AI Duet

Like the Continuator, other systems have used predictive models at the response stage to automatically accompany or embellish the user's performance. An interesting recent development has been to apply a deep RNN model in such a design. The Magenta project's AI Duet [27] 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 with the RNN model.

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. Although this may suggest that the RNN model has less relation to the user's particular performance than Continuator, as the RNN's memory state is conditioned by the user's notes, it is able to pick up stylistic information such as rhythm and harmony in practice. In fact, the RNN's ability to learn from a very large corpus of data can be a significant advantage; in the case of a novice user, the musical input may be very simple, and the RNN might be able to encourage or inspire them with new musical ideas.

4.3.3 The Neural Touchscreen Ensemble

The Neural Touchscreen Ensemble [29] is an RNN-driven simulation of a touchscreen ensemble experience. With this system, one human performer plays freely improvised music on an iPad's touchscreen and an ensemble performance is continually played back on three RNN-controlled iPads in response. In this case, both the human and computer-controlled iPads use a simple app that allows struck or sustained sounds from a limited selection of notes. A server tracks the human performance as a sequence of simple gestural patterns and, using an RNN, predicts an appropriate gestural response from three other members of a touchscreen ensemble. Touchscreen control signals matching these desired gestures are then sent to the three other iPads for performance. This system thus provides a simulation of an ensemble experience for a single touchscreen performer.

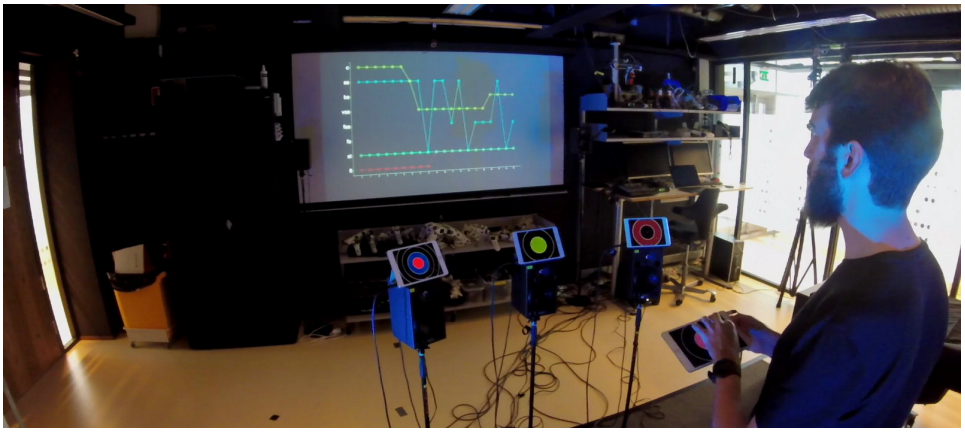

Figure 10: The neural touchscreen ensemble uses an RNN to predict ensemble responses to a human performer's gestures. The system supports quartet performances with three ANN-controlled iPads responding to one human performer.

In this system, the human performer continually produces a sequence of gestural symbols, recognised by the server, once each second. These symbols come from a simple vocabulary of 9 touchscreen gestures described in previous research on iPad ensemble performance [28]. Since these gestural signals could be considered as high-level musical notation, we consider them to occur at the response stage of the interactive music system. The temporal RNN model is configured to predict entries in three parallel sequences, that of the three ensemble performers. This gesture-RNN takes four gestures as input–the human performer's present gesture, and the ensemble's gesture at the last time step–and outputs the three gestures for the ensemble at the present step, this arrangement is illustrated in Figure 11. To simplify the implementation, these groups of four and three gestures are encoded as a

single integer, preserving the ordering of the ensemble members. The RNN design is thus similar to a CharRNN, with categorical input and output and featuring three layers of LSTM units.

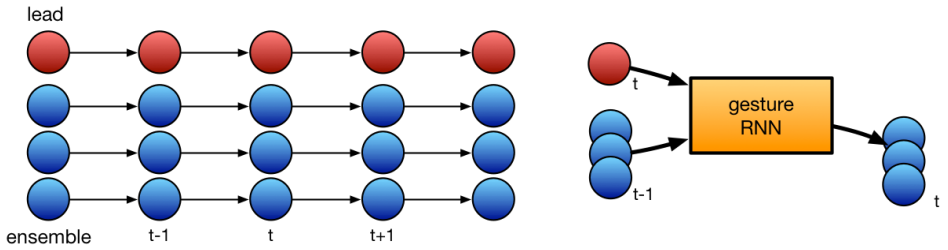

Figure 11: Gesture-RNN predicts appropriate gestural motions for three ensemble members based on present information about the human performer, and past information about the ensemble.

In the Neural Touchscreen Ensemble, the deep predictive model does not forecast future gestures for the ensemble as the predictions relate to the present time step. This model is, however, temporal, as it relies on input from the previous time-step and the memory state of its LSTM cells to make predictions. In terms of our framework, the predictive model takes data from the response stage of the human performer, and uses the predictions at the sensing stage of the ensemble. In this case, the RNN's predictions are in the form of high-level gestural signals, and a separate touch synthesis system is required to generate sensing signals for the ensemble iPads. In a more advanced system, the predictions could be in the form of touchscreen interactions, as with RoboJam's model.

It should be noted that performance accompaniment systems do not necessarily require deep learning to function. Biles' *GenJam* system [4], for instance, uses genetic algorithms to generate appropriate jazz-style accompaniments. Other systems using style-appropriate musical heuristics are often embedded into commercial music software. In the case of the Neural Touchscreen Ensemble, the musical content–freely improvised touch interaction–would not be easily described by standard music theory. A data-driven approach to modelling this kind of interaction was required.

5 Conclusion

In this paper, we have defined a framework for including predictive models in interactive music systems following a traditionally applied sense-process-response model, and we have reviewed how existing systems, including two from our own group, have implemented predictive models. A variety of ML techniques have been employed for the predictive models in these systems,

including both temporal and non-temporal models, those that forecast future values of a known time-series, and those that predict the present value of an unknown quantity. In each case, generating this unknown data has allowed these 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 generate ensemble responses from artificial agents.

Our review demonstrates that deep learning models, in particular, have much to offer predictive musical interaction. It's well accepted that RNN models have the capacity to learn long-range temporal dependencies and to learn from large corpora of training data. These 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; however, the input and output one-hot vectors actually encode multiple performer gestures. We suggest that other musical deep models could be incorporated into interactive music designs where new explorations of their music-making possibilities can be used in data-driven prediction.

Although we have discussed many interactive music systems that use predictive models, the role of predictions and how they fit into interaction is often not made explicit. We think that this undersells the importance of prediction in these systems and in musical performance in general. Embedding predictive intelligence into DMIs and NIMEs 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 understand how predictive models can be incorporated into these systems. In the final part of this paper we will discuss what we see as the benefits that predictive models can offer to interactive music designers and users.

5.1 Benefits of Prediction

5.1.1 Temporality

Music and sound are temporal phenomena and yet, the widespread framework for interactive music systems shown in Figure 5 does not necessarily consider the axis of time. Indeed, the fundamental archetype for interactive music systems is reactive; response necessarily follows gesture. We think that predictive models are vitally important to embedding a temporal axis into interactive music designs. In reality, predictive models *are* used in NIME and DMI design. Considering prediction to be an essential part of

interactive music design frameworks allows these temporal models to be properly understood, examined, and developed.

5.1.2 Proactivity

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 interactive music systems, 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 NIMEs; interfaces could change to afford upcoming musical needs as well as respond to the users' commands.

5.1.3 Adaptability

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, many such systems that are implemented with virtual reality, touchscreen, or haptic interfaces could be designed with flexible morphology that can adapt according to a predicted requirement. In a simple case, a touchscreen app might have easy and hard modes that are automatically selected based on a predictive model of the user's input. The BRAAHMS system has experimented with a similar interaction based on predicting cognitive state [54].

5.1.4 Generation

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 to introduce deep generative models into interactive music systems is that the musical data with new interfaces is often unknown and may not be well modelled by music theory. Predictive RNN models, such as that used in RoboJam, could be able to learn a wide variety of creative control data.

5.2 Final Remarks

Prediction has clear roles in musical performance. In this work we have shown how deep predictive models can fit into interactive music design and where they have been successfully implemented. 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 and NIME designs show strong use of multi-modal sensing, highly creative processing, and artistically savvy responses, temporal predictive models have been under-explored. We argue that predictive musical interaction, taking advantage of deep models such as RNNs, can have important benefits to users and performers. We offer this challenge to musical interface designers: Think temporally and let your instruments predict the future!

5.3 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.

5.4 Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric 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] 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.

[5] S.-J. Blakemore, D. M. Wolpert, and C. D. Frith. Central cancellation of self-produced tickle sensation. *Nature Neuroscience*, 1(7):635–640, 1998. doi:10.1038/2870.

[6] 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/.

[7] 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.

[8] 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.

[9] B. Caramiaux and A. Tanaka. Machine learning of musical gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '13, pages 513–518, 2013. URL: http://nime.org/proceedings/2013/nime2013_84.pdf.

[10] 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.

[11] B. A. Clegg, G. J. DiGirolamo, and S. W. Keele. Sequence learning. *Trends in Cognitive Sciences*, 2(8):275–281, 2017/11/28 1998. URL: http://dx.doi.org/10.1016/S1364-6613(98)01202-9, doi:10.1016/S1364-6613(98)01202-9.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] 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`.

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

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

[23] 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.

[24] 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.

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

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

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

[28] 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.

[29] 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.

[30] 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.

[31] C. P. Martin and J. Torresen. RoboJam: A musical mixture density network for collaborative touchscreen interaction. *ArXiv e-prints*, Nov 2017. ArXiv e-prints arXiv:1711.10746 [cs.HC]. URL: http://arxiv.org/abs/1711.10746.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] M. Puckette. Something digital. *Computer Music Journal*, 15(4):65–69, 1991. doi:10.2307/3681075.

[38] 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.

[39] 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/.

[40] 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.

[41] R. K. Sawyer. Group creativity: Musical performance and collaboration. *Psychology of Music*, 34(2):148–165, 2006. doi:10.1177/0305735606061850.

[42] 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.

[43] G. Schillaci, V. V. Hafner, and B. Lara. Exploration behaviors, body representations, and simulation processes for the development of cognition in artificial agents. *Frontiers in Robotics and AI*, 3:39, 2016. doi:10.3389/frobt.2016.00039.

[44] 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.

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

[46] 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.

[47] 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.

[48] A. Tidemann. An artificial intelligence architecture for musical expressiveness that learns by imitation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, NIME '11, pages 268–271, 2011.

[49] 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. URL: http://dx.doi.org/10.1007/978-3-642-04380-2_14, doi:10.1007/978-3-642-04380-2_14.

[50] 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.

[51] 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.

[52] B. Webb. Neural mechanisms for prediction: Do insects have forward models?, 2004. doi:10.1016/j.tins.2004.03.004.

[53] D. M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):338–347, 1998. doi:10.1016/S1364-6613(98)01221-2.

[54] 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.