# USING CONDITIONAL RESTRICTED BOLTZMANN MACHINES TO GENERATE TIMBRAL MUSIC COMPOSITION SYSTEMS

BY

MICHAEL J JUNOKAS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Informatics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Guy E Garnett, Chair
Associate Professor Paris Smaragdis
Professor Heinrich Taube
Associate Professor John Toenjes

# ABSTRACT

Machine-learning models have been successfully applied to musical composition in a variety of forms, including audio classification, recognition, and synthesis. The capability of algorithms to learn complex musical elements allows composers to more deeply investigate the development of their aesthetic. Coupled with the history of interdisciplinary solutions found in computer music and system aesthetics, this capability has led to an exploration of the integration of machine learning and music composition. Composition systems that take advantage of this integration have the opportunity to be connected with algorithms in theory, application, and art.

In my systems, conditional restricted Boltzmann machines (CRBM) synthesize musical timbre by learning autoregressive connections between the current output, an abstracted non-linear hidden feature layer, and past outputs. This provides a creative space where composers can synthesize audio spectra in collaboration with machines, defining novel creative systems that explore compositional material in an abstract, non-linear paradigm.

By implementing CRBMs in timbral-synthesis composition systems, I provide concrete support that such an integration advances art through the exploration of machine learning. I demonstrate this in a variety of audio synthesis experiments validating the capabilities of two algorithmic structures to synthesize and control timbre: a single layer conditional restricted Boltzmann machine (CRBM) and a single layer factored conditional restricted Boltzmann machine (FCRBM). I start by accurately synthesizing specific instrumental timbres and different musical pitches, demonstrating the aural capabilities of directly using the algorithms. I then build from these experiments, creating a set of compositional utilities that provide the composer with a rich pallet to provoke aesthetic introspection. These compositional utilities are then implemented in two music composition systems that synthesize and control timbre in application, where the algorithms themselves are designed and manipulated as a means to realize artwork.

Through the creation of music composition systems that are able to accurately synthesize and control musical timbre, I demonstrate these models have the capability of provoking the aesthetic introspection of composers. The resulting systems show the power and potential of integrating music composition and machine learning, endorsing an interdisciplinary approach to the development of art and technology.

*For Anne, Jack, and Charlie*

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to:

Professor Guy Garnett for his guidance and enthusiasm, that has made this work possible.

Professor Paris Smaragdis, Professor Henrich Taube, and Professor John Toenjes for their invaluable encouragement and criticism.

Doctor Donna Cox and the AVL team at NCSA for their unwavering support and creating an academic home essential for my research.

Karin Readel for her patience, diligence, and dedication.

Professor Robb Lindgren and the entire ELASTIC$^3$S group for providing a research space that nurtured my intellectual curiosity and confidence.

Kyungho Lee, Mohammad Amanzadeh, Yishuo Liu, and Shannon Cuykendall for motivating me to work better and aspire to ideals set forth in our collaborations.

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 Research Issues and Concept

Machine-learning applications in music have opened music composition to new methods of creativity. Sophisticated creative processes previously relegated to the composer such as material generation, defining complex mapping schemas, and managing higher order control of musical parameters can be modeled effectively by algorithms. This has led to a liberation of technological expression, expanding composers' musical aesthetics while driving algorithmic design in order to achieve that expression. These machine-learning music composition applications are a natural progression of the interdisciplinary precedent set by system aesthetics and computer music. These fields have leveraged the mutual development of creativity and technology, creating systems that expand the capabilities of participating agents through their integration.

This technological liberation based in integrated research generates opportunities to create music composition systems that develop aesthetics through the exploration and application of technology. These systems would challenge composers and their aesthetic approach, enriching composition in previously inaccessible ways, while defining algorithmic designs to meet their artistic needs. For composers to take advantage of such systems, implementations need to be able to utilize their compositional capabilities effectively, efficiently, and in application. I demonstrate an approach to developing integrated systems that realize the potential of these opportunities, creating human-machine music composition systems for timbral synthesis and control.

Timbral analysis and the digital manipulation of sound quality has allowed composers to delve deeply into their relationship with audio. From a data perspective, timbre is a highly dimensional, dynamic time series that is broadly diverse across sonic spectra. Defining generalized systems that can synthesize and control this space requires algorithms that can learn a flexible, yet robust set of parameters for creative interaction.

In order to develop systems that would be able to achieve complex timbral synthesis and control, I implemented conditional restricted Boltzmann machines (CRBM) and factored conditional restricted Boltzmann machines (FCRBM). CRBMs incorporate temporal elements into synthesis, using autoregressive relationships between past and current outputs to dynamically generate time-related material. CRBMs are also capable of mapping highly dimensional feature spaces into an organization of sigmoidal "energy" units, providing a new perspective of the data and its underlying patterns. FCRBMs extend the CRBM framework through the factorization of the algorithmic interactions, providing more developed connections within the model and providing composers with a method to direct synthesis externally. Musically, this is realized through the deconstruction of timbral elements, defining spectral features of sound at higher levels that can be used to synthesize, control, or transform timbres.

While CRBMs and FCRBMs have been applied and developed in a variety of research domains, the use of these algorithms in aesthetically concerned realms, such as music composition, remains largely unexplored. This motivated me to develop timbral music composition systems that leverage the unique capabilities of these algorithms, extending their use through application. By doing so, I give composers the ability to synthesize and control timbre, serving practical functionality in traditional synthesis applications and empowering aesthetic explorations into machine-driven creativity.

The conceptual framework of this research is grounded in an aesthetic perspective, focusing on enriching the creativity of artists through their interaction with systems. Starting from the fundamental work of machine-learning scientists, principally Graham Taylor and Geoffery Hinton, this perspective merges motivation from the theory of system aesthetics defined by J. W. Burnham, the art of Hans Haacke and Lisa Jevbratt, and the work of composers Iannis Xenakis, Kaija Saariaho, John Bischoff, and David Tudor. From this conceptual base, I develop music composition systems that connect humans and machines through the synthesis and control of dynamic timbres, integrating the unique capabilities of CRBMs and FCRBMs with the insight learned from these contexts.

I demonstrate the capabilities of these algorithms in a series of traditional synthesis and classification experiments. I test the ability to accurately synthesize different instrumental timbres and pitches using two different algorithmic setups: several single layer CRBMs, testing the fundamental capability of incorporating CRBMs with timbral synthesis; and a single layer FCRBM, testing the capability of processing more complex abstract structures that are user tunable. I show that CRBMs are particularly effective in these traditional synthesis tasks, achieving accuracies of over 94% in the classification of synthesized material. I also show that FCRBMs are effective in modeling instrumental pitches, achieving accuracies of over 96% in classification experiments, while additionally generating transitional material between classes that was not specifically defined in the data.

From the accurate synthesis within these initial experiments, I then design models to accomplish specific musical tasks, developing a series of compositional utilities. These include manipulating the dynamic envelope of different timbres, generating unique and dynamic soundscapes from primary source material, and performing synthesis via stylistic label manipulation of the algorithms.

From the initial experiments and the resulting compositional utilities, I realize two timbral music composition systems, demonstrating the potential of integrating algorithmic design with musical creation. I show the process of composing from an abstract formal construct, using a network of CRBMs and FCRBMs to translate dance choreography to timbral synthesis in the collaborative work *a performer's perspective*(2017). I show the capability of CRBMs and FCRBMs to digitally synthesize and control a variety of dynamic timbral textures from a limited audio vocabulary through multimodal interaction in the immersive art installation series *is That('s) all there is*(2016-2017).

Through the creation of these timbral music composition systems, composers are able to develop more complex relationships between their use of technology and their aesthetic. These more developed relationships challenge composers to investigate their process, reconsider their interactions with technology, and redefine their perspectives on artistic control. By redefining their compositional approach and delegating complex timbral synthesis and control to algorithms, composers can channel their technological concerns toward aesthetic rather than logistic and technical proficiency, potentially developing a new form of interaction that requires a different understanding of their role in musical creation. These music composition systems expand the composer's aesthetic through machine-interaction, leading composers to embrace the capabilities of such systems, bringing a more developed approach to music technology, and integrating computational models with musical aesthetics.

To meet this approach, algorithms must be continually developed and designed to adequately address such complex understandings. The implementation of CRBMs and FCRBMs in these systems provides a foundation for building more complex models (e.g. deep belief nets) that further stretch creative paradigms.

Through the demands of these music composition systems, technology is used to address the complexity of timbral synthesis and control, providing solutions that drive the design and application of CRBMs and FCRBMs and enrich the aesthetic investigation of composers. By developing music composition systems that use these models effectively and efficiently in application, I provide a compositional methodology that incorporates algorithms directly into creativity, generating an approach to music composition that integrates machine learning and art to advance aesthetic and computational concerns of composers.

## 1.2 Outline of Dissertation

**Chapter 2** provides an overview of the related research in applied machine-learning and the supporting theory that led to the technical construction and design of the CRBMs and the FCRBMs in my music composition systems, showing how they are ideal algorithms to achieve timbral synthesis and control.

**Chapter 3** provides an overview of the related work in art theory that motivates the integration of technological systems with aesthetics. Specifically, I explore the theory of system aesthetics through the curation of J.W. Burnham's *Software* exhibition, the early systems work of Hans Haacke, and the application of systems theory to Lisa Jevbratt *Interspecies Collaborations.* I connect this system perspective to music, investigating formal (Xenakis), timbal (Saariaho), and performative (Bischoff, Tudor) uses of timbral synthesis and control in music composition.

**Chapter 4** demonstrates the capabilities of CRBMs and FCRBMs to achieve complex timbral tasks. I articulate the specifications for the applied algorithms in terms of data, model parameters, and general framework. I then test the algorithms in a series of different experiments including two traditional synthesis tasks (synthesizing different instruments from the same musical family and different pitches from the same instrument) and the creation of three sets of compositional utilities (manipulating dynamic envelopes, sustaining non-repeating generative timbres, synthesis and filtering using composer chosen labels). From the results of these experiments, I show the potential of using and designing these algorithms for musical composition.

**Chapter 5** describes the implementation of two music composition systems that utilize CRBMs and FCRBMs in the artworks *a performer's perspective*(2017) and *is That('s) all there is*(2016-2017), realizing the capability of these algorithms to create complex, timbral synthesis and control systems that are defined by the integration of machine learning and music composition.

**Chapter 6** concludes by summarizing the findings of this research and discussing future directions for the integration of music composition with system design and the development of these algorithms. I present the unique advantages demonstrated in the validation tasks, compositional utilities, and creative work of this dissertation. I explore the potential contributions of this research in musical and computational domains, pointing to specific growth opportunities that result from the integration of system aesthetics, music composition, and algorithmic application to address creative concerns.

# CHAPTER 2

# BACKGROUND AND RELATED WORK IN MACHINE LEARNING

In this chapter I describe the related work being done with non-linear, hidden layer, machine-learning algorithms, specifically focusing on applications in the audio domain and my choice of CRBMs and FCRBMs as the foundational algorithms for my timbral music composition systems. In section 2.1, I discuss previous research done in machine learning and applied data-synthesis fields. In section 2.2, I describe the theoretical underpinnings of the CRBM and the FCRBM, outlining why they are ideal algorithms for timbral synthesis and control in music composition systems.

## 2.1 Background and Related Work in Machine Learning

An artificial neural network is a statistical learning model that learns non-linear representations of datasets, generating an approximate function that generalizes mappings of an input space to an output space [1]. The use of such a model, that develops connections directly from the data, opens composers to algorithmic patterns untethered by human design. This machine-driven vocabulary gives growth to new forms and explorations, not limited by human conceived solutions, providing a fertile ground for expanding creativity. Non-linear representations have been used in several different audio technology applications such as gestural-audio interfaces [2] [3] [4], cybernetic musical systems [5] [6], and musical parameter control [7] [8].

From this expanded vocabulary of machine-derived connections, composers need ways to control mappings and outputs in order to facilitate their own creativity effectively. The theoretical construction of these non-linear models provides a scaffold from which composers can process material to fit their needs. The models learn hidden layers, an abstracted representation of the data consisting of a number of hidden units. Each hidden unit of the layer activates according to an applied energy model [9], as is the case with the generative algorithm, the conditional restricted Boltzmann machine (CRBM). The hidden layer of the CRBM gives the composer a way to interact with the algorithms self-defined connections and the resulting data it synthesizes. Learned abstractions provide a creative ground directly related to the data, yet structured in a new, non-linear form, giving a different perspective from which to explore aesthetic concerns. Investigating musical organization and structure through technological applications has been a vital element of several composers' process (see 3.2.3) and a chief aim of my research. In CRBMs, this abstract hidden space is affected by past frames of the data, with material being grounded in autoregressive connections. These temporal connections give CRBMs a direct correlation to time-series data, such as musical timbre, and provide an entry to controlling non-linear models.

In CRBMs, the algorithm learns the connections/weights and respective biases of the hidden layers using contrastive divergence (CD) to approximate the maximum-likelihood function in a tractable manner [10] [11]. The tractability of this model is essential to processing and working with hugely dimensional, time-series datasets practically, provoking deeper explorations that would not be possible without this efficiency. Due to this capability, CRBMs have been used successfully applied in several machine-learning tasks such as handwritten digit recognition and generation [12] and facial recognition [13].

From CRBMs, a variety of algorithmic extensions have been developed, such as Graham Taylor's factored conditional restricted Boltzmann machine (FCRBM) [14]. A FCRBM is a CRBM where intermediate layers of 'factors' are used to model the interactions between the internal parameters (i.e. the visible, hidden, and feature units) of the algorithm. The inclusion of factorization reintroduces a layer of human agency into the algorithms, learning macro-stylistic tendencies that composers can tune during synthesis, similar to the digital timbral synthesis and manipulation processes of past composers (see 3.2.2). This gives composers a method of directing the algorithms without restricting the expanded vocabulary generated by the algorithm.

The combinatorial nature of CRBMs and FCRBMs lend particularly well to musical applications, specifically compositional perspectives. The autoregressive nature of the algorithms place its modeling and synthesis in direct parallel with the temporal environment of music. The models learn several weights and biases in combination to create an appealing architecture of horizontal (i.e. autoregressive/temporal) and vertical (i.e. hidden layer) connections, providing composers with multiple levels to structure and formalize their process and compositions.

Through the incorporation of the FCRBMs' user-controlled factors, composers are provided with an additional element of high-level control, giving direct access to the connections between the current output and the internal parameters of the algorithm. This high level structure can be linked to formal applications, as done in past music compositions (see 3.2.1), as well as provide access to previously unexplored perspectives of the musical data through the algorithm's architecture. The augmentation of these formal aspects into a dynamic generative model provide the composer with a higher structural control of a vocabulary of micro-consequences, incorporating algorithms directly into formal compositional considerations.

While the majority of CRBM and FCRBM applications have been focused in recognition tasks, models that synthesize user-defined 'styles,' have emerged [14]. My research delves deeper into these algorithms as a means of facilitating material synthesis, digital control, and creative expansion. This extended look into the theoretical construction of CRBMs and FCRBMs highlight the opportunities to link these algorithmic structures to compositional utility and expression.

## 2.2 Theoretical Construction of the CRBM and FCRBM

CRBMs and FCRBMs are energy-based models that learn weights and biases between data and user-defined elements, measuring the difference between the expectation of the data generated from the training set, $E[\theta_{data}]$, and the expectation of the reconstruction of the data generated using a Gibbs sampling algorithm (see 2.8), $E[\theta_{recon}]$. By measuring the difference between the data and its reconstruction, the model is able to iteratively learn abstract weights and biases that can be used to represent and synthesize new data.

In order to generate the $E[\theta_{data}]$, samples from the training data are used as the visible units (i.e. inputs) of the algorithm. These visible units are used to determine the activations in the abstract hidden unit layer, using the bias of the hidden units plus the sum of the weight matrix multiplied with the visible units, run through a sigmoid function, to determine if the hidden unit is 'on' ($h_j = 1$) or 'off' ($h_j = 0$) (explained and specifically shown in the context of RBMs in equation 2.6).

In order to generate the $E[\theta_{recon}]$, visible units are reconstructed using the bias of the visible units plus the sum of the weight matrix multiplied with the hidden units generated by the $E[\theta_{data}]$ step. The hidden units are determined from these reconstructions, using the same process as described in the $E[\theta_{data}]$ step (explained and specifically shown related to RBMs in equation 2.8).

The difference of these expectations can then be used to update the weights and biases of the model (i.e. RBM, CRBM, or FCRBM), iterating for a predetermined number of epochs or until an error function reaches a suitably low threshold. Within each iteration of the difference, the estimation using Markov chain Monte Carlo (MCMC) sampling will eventually converge to a stationary distribution, thus we can expect the gradient to eventually converge to zero. When the weights and biases are small, the MCMC converges rapidly and we can approximate the gradient for the parameters.

Specifically, in the context of CRBMs and FCRBMs, the restricted nature of the algorithms (i.e. no visible-visible and hidden-hidden connections, only visible-hidden connections) allows for the model to use contrastive divergence (CD) for learning the gradient due to the conditional independence of the visible units with respect to each other and the conditional independence of the hidden units with respect to each other. This provides an efficient and tractable method for calculating the gradient of the weights and biases of RBMs. The FCRBM is an extension of this framework, expanding the energy based model of CRBM to incorporate stylistic, context-sensitive factors into inference and learning.

### 2.2.1 Energy-based Model Theory

Energy-based models [15] define a model's probability distribution through the energy function:

$$p(x) = \frac{e^{-E(x)}}{Z} \tag{2.1}$$

where Z is a normalization, partition function represented by:

$$Z = \sum_x e^{-E(x)} \tag{2.2}$$

In a BM [16], the numerator can be defined as a probabilistic distribution modeling the visible and hidden units, thus making the distribution:

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{2.3}$$

where the energy function $E(v, h)$ becomes:

$$E(v, h) = -\sum_{ij} W_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j \tag{2.4}$$

The partition function $Z$ becomes intractable to compute, as it becomes the sum over all possible joint probabilities:

$$Z = \sum_{v,h} E(v, h) \tag{2.5}$$

### 2.2.2 Restricted Boltzmann Machine (RBM) Theory

A RBM [17] makes the partition function tractable by removing interactions between hidden units, creating an estimation of the negative gradient based on a fixed number of model samples, with connections that only exist between the visible and hidden units. Due to the structure of RBMs, conditional independence exists between hidden and visible units. Thus, a sample of the negative distribution $v, h$ can be estimated by attaching visible units to a training vector and sampling the hidden units in parallel according to:

$$p(h_j = 1|v) = \frac{1}{1 + e^{-b_j - \sum_i W_{ij} v_i}} \tag{2.6}$$

where the visible-hidden weight with the added hidden bias is run through a logistic sigmoid function, generally represented as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

Hidden units are then 'activated' (i.e. $h_j = 1$), if the value is greater than a random value. The corresponding positive distribution is calculated by performing alternation Gibbs sampling, iterating between $p(h|v)$ and:

$$p(v_i = 1|h) = \frac{1}{1 + e^{-a_j - \sum_j W_{ij} h_j}} \tag{2.8}$$

Performing this sampling using contrastive divergence[11] gives us the learning updates of:

$$\Delta W_{ij} \propto \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \tag{2.9}$$

$$\Delta b_j \propto \langle h_j \rangle_{data} - \langle h_j \rangle_{recon} \tag{2.10}$$

$$\Delta a_i \propto \langle v_i \rangle_{data} - \langle v_i \rangle_{recon} \tag{2.11}$$

The ability to learn an abstract representation of the data, (i.e. a set of binary hidden units), provides a new perspective for composers. By learning the connections between an abstract space (i.e. hidden units) and an audible space (i.e. visible units), composers can manipulate, generalize, and organize their aesthetic direction through an algorithmically defined paradigm. This provides an approach that translates densely complex data such as timbre into a more approachable form (as applied in 4.4.2). Like the artists exploring systematic approaches to aesthetics (as detailed in 3.1), the RBM provides an systemized approach to the synthesis and control of highly dimensional artistic data.

### 2.2.3 Conditional Restricted Boltzmann Machine (CRBM) Theory

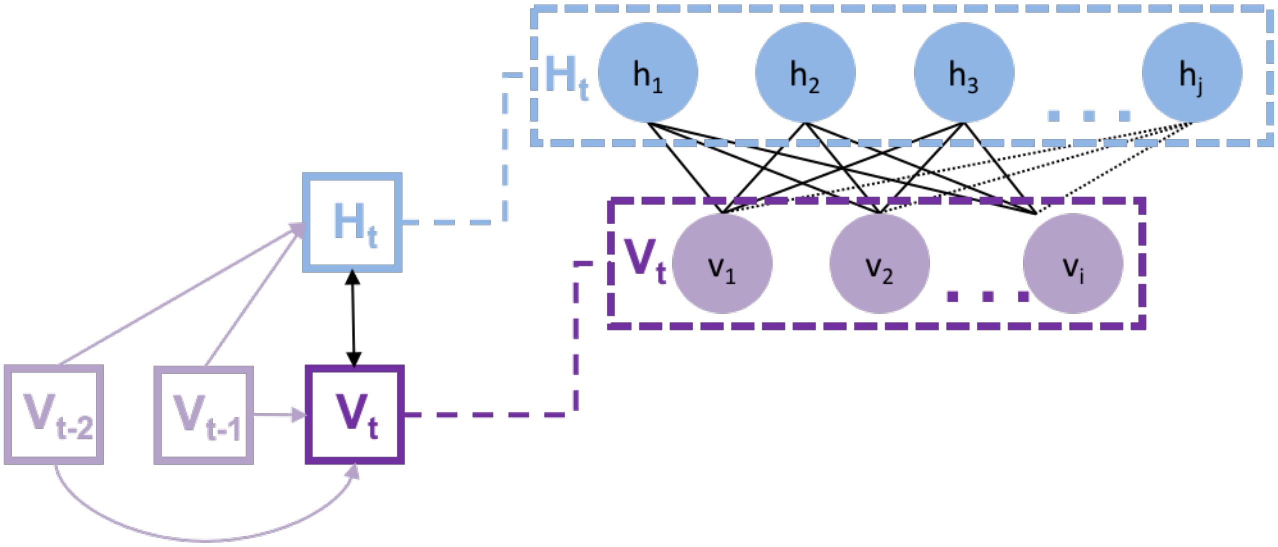


Figure 2.1: General architecture of a second order CRBM

While the RBM provides an abstracted space from which to create, generalize, and explore the data, it only learns connections between current hidden units and current visible units. In a CRBM [13] [18], temporal information is incorporated into the model by adding previous frames of the data as additional fixed inputs, making autoregressive connections to current visible and hidden units. This places the model in time, learning connections and generating material with respect to the past. This autoregressive connection provides a concrete parallel to musical considerations (see 3.2), adapting synthesis and control to what has come before. Weights and biases learned with this consideration provide composers with a structure that is based on temporal relationships, giving access to the dynamic aspects of the algorithms that are based concretely in time (see 4 for examples of experiments applying these temporal considerations).

Transitioning from a RBM to a CRBM results in the energy function:

$$E(v_t, h_t | v_{<t}) = \sum_i v_i \hat{a}_{i,t} - \sum_{ij} W_{ij} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_j \qquad (2.12)$$

where given the current $(v_t)$ and past frames $(v_{<t})$, the negative distribu-

tion over hidden units becomes:

$$p(h_{j,t} = 1|v_t, v_{<t}) = \frac{1}{1 + e^{-\hat{b}_{j,t} - \sum_i W_{ij} v_{i,t}}} \tag{2.13}$$

where $\hat{b}_{j,t}$ is now:

$$\hat{b}_{j,t} = b_j + \sum_k B_{kj} v_{k,<t} \tag{2.14}$$

and the positive distributed reconstruction over visible units becomes:

$$p(v_{i,t}|h_t, v_{<t}) = \hat{a}_{i,t} + \sum_j W_{ij} h_{j,t} \tag{2.15}$$

where $\hat{a}_{i,t}$ is now:

$$\hat{a}_{i,t} = a_i + \sum_k A_{ki} v_{k,<t} \tag{2.16}$$

Now that multiple frames are incorporated into the model, the respective updates are summed over all time steps and gives us the learning updates of:

$$\Delta W_{ij} \propto \sum_t (\langle v_{i,t} h_{j,t} \rangle_{data} - \langle v_{i,t} h_{j,t} \rangle_{recon}) \tag{2.17}$$

$$\Delta A_{ki} \propto \sum_t (\langle v_{i,t} v_{k,<t} \rangle_{data} - \langle v_{i,t} v_{k,<t} \rangle_{recon}) \tag{2.18}$$

$$\Delta B_{kj} \propto \sum_t (\langle h_{j,t} v_{k,<t} \rangle_{data} - \langle h_{j,t} v_{k,<t} \rangle_{recon}) \tag{2.19}$$

$$\Delta a_i \propto \sum_t (\langle v_{i,t} \rangle_{data} - \langle v_{i,t} \rangle_{recon}) \tag{2.20}$$

$$\Delta b_i \propto \sum_t (\langle h_{j,t} \rangle_{data} - \langle h_{j,t} \rangle_{recon}) \tag{2.21}$$

where k is the number of steps for the reconstruction distribution, using the training data for the visible units.

Generating audible data with respect to an abstracted hidden layer that is also affected by the past demonstrates a more apparent correlation to the musical forms of several systematic composers (see 3.2.3). In the realization of music, sonic objects are not created to be in isolation but as a series of relational consequences within time, apparent at micro (e.g. dynamic envelope construction, as explored in 4.4.1) and macro (e.g. composer control-mappings or formal compositional construction, as explored in 5) scales. The CRBM takes the theory of the RBM and places it in time, making it much more powerful and appealing to temporally concerned composers.

### 2.2.4 Factored Conditional Restricted Boltzmann Machine (FCRBM) Theory

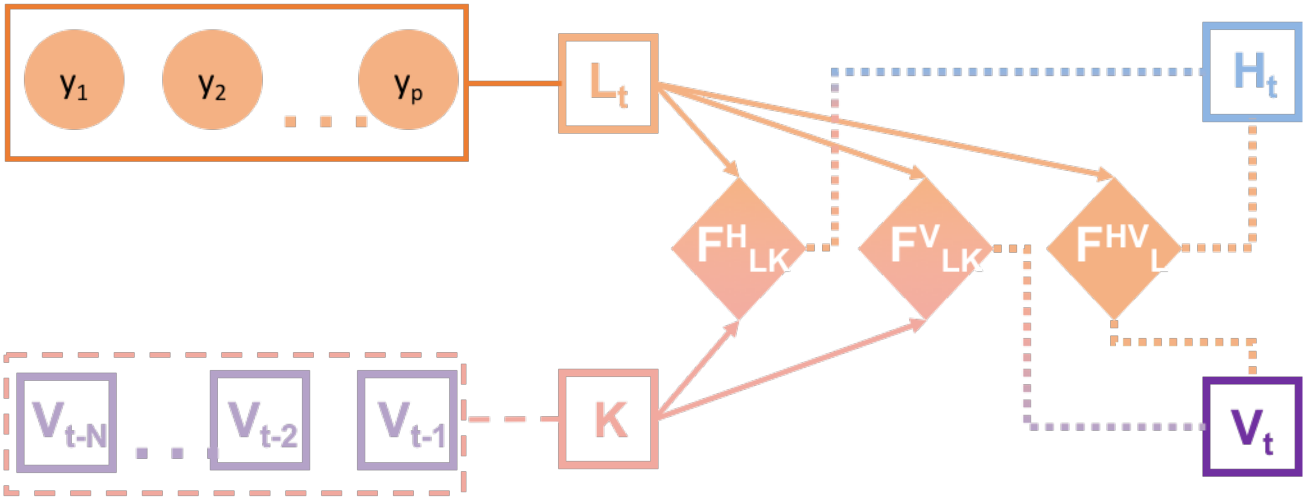


Figure 2.2: General architecture of a FCRBM

In a FCRBM [14], an additional set of deterministic 'factors' are introduced. These factors learn the interactions between the internal parameters of the model (i.e. visible, hidden, past visible, and feature units), providing the composer with a way to directly manipulate the algorithm through external labels. This empowers composers to direct the algorithm as a fully integrated component of the system, creating a relationship much more analogous to collaborative opportunity rather than composer-centric limitations (see 3.1 for art theory basis).

By factorizing the algorithm, the resulting energy function becomes:

$$E(v_t, h_t|v_{<t}) = \sum_i v_i \hat{a}_{i,t} - \sum_f \sum_{ij} W^v_{if} W^h_{jf} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_j \tag{2.22}$$

where the biases are also factored, making $\hat{a}_t$ and $\hat{b}_t$:

$$\hat{a}_{i,t} = a_i + \sum_m \sum_k A^v_{im} A^{v<t}_{km} v_{k,<t} \tag{2.23}$$

$$\hat{b}_{j,t} = b_j + \sum_n \sum_k B^h_{jn} B^{v<t}_{kn} v_{k,<t} \tag{2.24}$$

In these biases, $m$ and $n$ relate to the factoring of direct connections $A$ and $B$.

With the model now factored, features controlled by labels are introduced, allowing for a 'stylistic' control of the interactions within the model, as applied in [19], [20], and [14]. These labels act as a channel of interaction between composer and algorithm, providing an aesthetic access point for cohesive, theoretically established data synthesis (see 5.1 and 5.2).

With the introduction of feature-label term $z_{l,t}$ and the replacement of $W_{i,j}$ with three different weight matrices representing visible ($W^v_{if}$) , hidden ($W^h_{jf}$), and feature ($W^z_{lf}$) interactions, the energy function becomes:

$$E(v_t, h_t | v_{<t}, y_t) = \sum_i v_i \hat{a}_{i,t} - \sum_f \sum_{ijl} W^v_{if} W^h_{jf} W^z_{lf} v_{i,t} h_{j,t} z_{l,t} - \sum_j \hat{b}_{j,t} h_j \tag{2.25}$$

and the corresponding biases become:

$$\hat{a}_{i,t} = a_i + \sum_m \sum_{kl} A^v_{im} A^{v<t}_{km} A^z_{lm} v_{k,<t} z_{l,t} \tag{2.26}$$

$$\hat{b}_{j,t} = b_j + \sum_n \sum_{kl} B^v_{jn} B^{v<t}_{kn} B^z_{ln} v_{k,<t} z_{l,t} \tag{2.27}$$

In order to learn the hidden unit activations in the model with these label-feature interactions, the negative distribution over the hidden units becomes:

$$p(h_{j,t} = 1 | v_t, v_{<t}, y_t) = \frac{1}{1 + e^{-\hat{b}_{j,t} - \sum_f W^h_{jf} \sum_f \sum_i W^v_{ij} v_{i,t} \sum_l W^z_{lf} z_{l,t}}} \tag{2.28}$$

and the positive distributed reconstruction over visible units becomes:

$$p(v_{i,t} | h_t, v_{<t}, y_t) = \hat{a}_{i,t} + \sum_f W^v_{if} \sum_j W^h_{jf} h_{j,t} \sum_l W^z_{lf} z_{l,t} \tag{2.29}$$

The inclusion of the three term interactions changes the gradient update rules to products that involve "the activity of the respective unit, and the total input to the factor from each of the two other sets of units involved in the three-way relationship [14]," which gives us the learning updates of:

$$\Delta W^v_{if} \propto \sum_t \left( \left\langle v_{i,t} \sum_j W^h_{jf} h_{j,t} \sum_l W^z_{lf} z_{l,t} \right\rangle_{data} - \left\langle v_{i,t} \sum_j W^h_{jf} h_{j,t} \sum_l W^z_{lf} z_{l,t} \right\rangle_{recon} \right) \tag{2.30}$$

$$\Delta W^h_{jf} \propto \sum_t \left( \left\langle h_{j,t} \sum_i W^v_{if} v_{i,t} \sum_l W^z_{lf} z_{l,t} \right\rangle_{data} - \left\langle h_{j,t} \sum_i W^v_{if} v_{i,t} \sum_l W^z_{lf} z_{l,t} \right\rangle_{recon} \right) \tag{2.31}$$

$$\Delta W^z_{lf} \propto \sum_t \left( \left\langle z_{l,t} \sum_i W^v_{if} v_{i,t} \sum_l W^h_{jf} h_{j,t} \right\rangle_{data} - \left\langle z_{l,t} \sum_i W^v_{if} v_{i,t} \sum_l W^h_{jf} h_{j,t} \right\rangle_{recon} \right) \tag{2.32}$$

$$\Delta A^v_{im} \propto \sum_t \left( \left\langle v_{i,t} \sum_k A^{v<t}_{km} v_{k,<t} \sum_l A^z_{lm} z_{l,t} \right\rangle_{data} - \left\langle v_{i,t} \sum_k A^{v<t}_{km} v_{k,<t} \sum_l A^z_{lm} z_{l,t} \right\rangle_{recon} \right) \tag{2.33}$$

$$\Delta A^{v<t}_{km} \propto \sum_t \left( \left\langle v_{k,<t} \sum_i A^v_{im} v_{i,t} \sum_l A^z_{lm} z_{l,t} \right\rangle_{data} - \left\langle v_{k,<t} \sum_i A^v_{im} v_{i,t} \sum_l A^z_{lm} z_{l,t} \right\rangle_{recon} \right) \tag{2.34}$$

$$\Delta A^z_{lm} \propto \sum_t \left( \left\langle z_{l,t} \sum_i A^v_{im} v_{i,t} \sum_l A^{v<t}_{km} v_{k,<t} \right\rangle_{data} - \left\langle z_{l,t} \sum_i A^v_{im} v_{i,t} \sum_l A^{v<t}_{km} v_{k,<t} \right\rangle_{recon} \right) \tag{2.35}$$

$$\Delta B^h_{jn} \propto \sum_t \left( \left\langle h_{j,t} \sum_k B^{v<t}_{kn} v_{k,<t} \sum_l B^z_{ln} z_{l,t} \right\rangle_{data} - \left\langle h_{j,t} \sum_k B^{v<t}_{kn} v_{k,<t} \sum_l B^z_{ln} z_{l,t} \right\rangle_{recon} \right) \tag{2.36}$$

$$\Delta B^{v<t}_{kn} \propto \sum_t \left( \left\langle v_{k,<t} \sum_j B^h_{jn} h_{j,t} \sum_l B^z_{ln} z_{l,t} \right\rangle_{data} - \left\langle v_{k,<t} \sum_j B^h_{jn} h_{j,t} \sum_l B^z_{ln} z_{l,t} \right\rangle_{recon} \right) \tag{2.37}$$

$$\Delta B^{z}_{kn} \propto \sum_t \left( \left\langle z_{l,t} \sum_j B^h_{jn} h_{j,t} \sum_k B^{v<t}_{kn} v_{k,<t} \right\rangle_{data} - \left\langle z_{l,t} \sum_j B^h_{jn} h_{j,t} \sum_k B^{v<t}_{kn} v_{k,<t} \right\rangle_{recon} \right) \tag{2.38}$$

Additionally, the weights connecting the labels to the features can be learned through back-propagation, with the gradients learned through CD [14]. Updating this weight, $R$ can be done with:

$$\Delta R_{pl} \propto \sum_t \left( \langle C_{l,t} y_{p,t} \rangle_{data} - \langle C_{l,t} y_{p,t} \rangle_{recon} \right) \tag{2.39}$$

where $C_{l,t}$ is

$$\Delta C_{l,t} = \sum_f W^z_{lf} \sum_j W^v_{if} v_{i,t} \sum_j W^h_{jf} h_{j,t} + \sum_m A^z_{lm} \sum_i A^v_{im} v_{i,t} \sum_k A^{v<t}_{km} v_{k,<t} + \sum_n B^z_{ln} \sum_j B^h_{jn} h_{j,t} \sum_k B^{v<t}_{kn} v_{k,<t} \tag{2.40}$$

The static hidden ($b_j$) and visible biases ($a_i$) are updated as in a regular CRBM:

$$\Delta a_i \propto \sum_t (\langle v_{i,t} \rangle_{data} - \langle v_{i,t} \rangle_{recon}) \tag{2.41}$$

$$\Delta b_i \propto \sum_t (\langle h_{j,t} \rangle_{data} - \langle h_{j,t} \rangle_{recon}) \tag{2.42}$$

As a result of these factor and feature additions, the FCRBM becomes an attractive algorithm for spectral synthesis and control. The time-based hidden layer realized in the CRBM can be modulated by the composer, allowing for the algorithm to be directed in synthesis without prescribing specific outcomes. These capabilities present unique possibilities for synthesis and control, generating material that is dynamic yet controlled within the confines of the composer's intent.

### 2.2.5 Synthesis from Trained Models

After the weights and biases of the algorithms are learned, synthesis occurs by initializing the model with sample data, defining model parameters, and iterating through a set number of Gibbs sampling steps, going through a forward pass of the trained model for the desired number of synthesized samples. For the FCRBM, this results in first calculating the constant during inference (a factored analog to the bias constant used in CRBMs) and constant during reconstruction based on the learned parameters using the equations 2.26 and 2.27.

Using these constants, synthesized data is generated the same way that the visible data is reconstructed in the learning steps of the algorithms. First, the hidden units are generated from the learned weights (as done in equation 2.28) given an initialization sample equal to the order of the model, iterating through a predetermined number of Gibbs sampling steps, and taking a mean-field approximation of those hidden units. With that mean-field approximation, the hidden units are multiplied by learned feature and factor matrices, and the reconstruction constant is added (as done in equation 2.29) to synthesize a frame of the data. Successive frames of data are then synthesized based on previously generated outputs.

## 2.3 Summary

In this chapter I described the foundational research that provides the basis for the algorithms used in my timbral music composition systems, specifically the CRBM and FCRBM energy models. From this established research, I outlined the theoretical construction of these algorithms, providing insight into the unique capabilities they posses that make them ideal algorithms for synthesizing timbre in music composition systems.

# CHAPTER 3

# BACKGROUND AND RELATED WORK IN AESTHETICS AND ART THEORY

In this chapter, I connect the machine-learning base described in chapter 2 with artistic work, showing the aesthetic and conceptual development of timbral music composition systems as a union of aesthetics and machine learning. In section 3.1, I give an overview of system aesthetics, describing its theoretical implications and the impact it has had on machine-driven creativity, citing the theoretical and curatorial work of J.W. Burnham, the art of Hans Haacke, and the work of Lisa Jevbratt. In section 3.2, I describe the impact of music composers, Iannis Xenakis, Kaija Saariaho, John Bischoff, and David Tudor, discussing how their work realizes systematic applications in music, and directly motivates the technological and creative decisions that define my research.

## 3.1 System Aesthetics

J.W. Burnham, in his work *Systems Esthetics* [21] and "The Aesthetics of Intelligent Systems [22]", provided the foundation for system aesthetics, merging the approach of artists and scientists of the 1970s with electronic information processing, creating "human enhancement through man-machine relationships [22]" and transitioning from an object-oriented culture to a systems-oriented culture, that is, from 'things' to "the way things are done [21]". The central objective of this enhancement was to be able to interact creatively with computational systems, developing "a dialogue where two systems gather and exchange information so as to change constantly the states of each other [21]." Burnham further presented the need for a shift in our relationship with machines:

> The continued evolution of both communications and control technology bodes a new type of aesthetic relationship, very differ-

> ent from the one-way communication of traditional art appreciation as we know it. A great deal of technological rationalization has derived from this attitude, which has led us to think in terms of human domination and environmental passivity. The change that I perceive, however, encourages the recognition of man as an integral of his environment.[22]

With this assessment of technology, Burnham outlines a central influence on my research: if artists investigate their role as "an integral" of the system rather than a dominant controller, more opportunities for considering aesthetic investigations result. By focusing on the creative interactions of human agents as part of a system (i.e. integrating machine agents, not dominating them), aesthetic concepts and creativity can grow beyond the one-way methodology of past artistic traditions. By allowing technology to enter as an active component in the creative process rather than a passive element, these types of systems incorporated the artist into the very environment from which they work, developing a completely different and novel creative paradigm.

Investigating the dynamics of this paradigm, through the parameter definition of a largely automated process, defining the structural boundaries of agent interactions, and designing a system architecture from an aesthetic perspective, all provide fruitful opportunities for creativity and are especially conducive to CRBM and FCRBM application. My research distinguishes itself from Burnham's foundational concepts by expanding the human-machine interactions, continuously developing from the exchange of information and resulting feedback. Structural and technological limitations of Burhnam's automated systems are circumvented by implementing CRBMs and FCRBMs, algorithms capable of reacting to user input, past history of input, and it's own abstract formalism. These multiple components of the algorithm provide new opportunities for technological development and system aesthetic exploration.

Burnham's vision was realized in the curation of *Software: Information Technology: Its New Meaning for Art*, an exhibition that highlighted a variety of technologies that focused on the "interaction between people and their electronic and electromechanical surroundings [23]." Works throughout the exhibition emphasized dynamic interactions between humans and machines:

*Composer* [24] empowered composers and users to manipulate any number of basic sonic elements of an audio synthesis system, creating an aural atmosphere of their own design; *The Conversationalist* [25] allowed people to record stories inspired by stochastically chosen wordsets; *Vision Substitution System* [26] coupled with *Light Pattern Box (Electrochrome)* [27] invited users to sit in a chair that had 400 vibrators mounted on its back, transforming camera data it into a tactile image and creating colored light patterns autonomously or in response to a viewer's rhythmic input.

### 3.1.1 Hans Haacke's Open Systems

Another artist Burnham included in the exhibition was Hans Haacke [28] [29] whose system work especially informs my research. Haacke's early work with open systems addressed concerns presented by incorporating algorithmic and human agents in successful collaboration. His *Photo-Electric Viewer-Programmed Coordinate System* [30] used a system of infrared light and photo-resistors to allow for spectator's to control lights placed above them in the space, defining the experience by the spectator's movement and attempt to discover its underlying logic.

While not as technologically oriented, Haacke's earlier exhibition at MIT in 1967 [31], displayed similar systematic approaches, contemplating non-human agency within systems. *Wave*(1965) invited observers to actively participate in a system, setting a large, sealed plastic container of water suspended from the ceiling in motion, thus animating the work directly through their involvement. *White Waving Line*(1967) used a long, thin piece of fabric, caught floating in a stream of air generated by a fan, creating an aerodynamic sculpture that was a result of Haacke's design but out of his direct control. *Condensation Cube*(1967), was a clear, sealed box is filled with a small amount of water, where physical systems and natural processes act upon each other within the human imposed structure.

This exploration of the underlying hidden structure of systems and using that exploration as a means to create art is directly linked with my investigation of the learned abstraction of algorithmically generated hidden layers of CRBMs and FCRBMs. Intuiting the base logic of this abstraction occurs through experience, interaction, and dialogue with the machine architecture.

In such an environment, the consequences, resulting outputs, and process of discovering these elements of the system contribute to its artistic value. Similarly, my research provides a fertile ground for the exploration of the system, specifically defining synthesis through external, composer-determined labels, allowing for the composer to interact directly with the algorithm and material it is synthesizing (see 5.2). Due to the non-linear nature of CRBMs and FCRBMs, the activation of different hidden units results in a variety of sonic consequences that are only realized through the exploration of that abstraction (see 4.4). With Haacke's open systems, problems are not addressed from a linear, direct perspective, but approached considering aspects and relationships within the system. Components derive meaning from their place and interaction within, very similar to the factorization of the FCRBM (see 2.2.4).

The relationships developed from the composer's interactions with other elements of the system are at the very essence of my research. Consequences of these interactions are subservient to further developing the relationships between the elements, through mutual investigation and protocol. With this emphasis on exploring relationships within an open system, composers are able to reshape their role, finding creativity as an elemental participant, exploring the various interactive contingencies between themselves and the algorithmic components of their compositional system. Specific to this work, the interactions of the composer with the algorithms becomes factorized, resulting in timbral synthesis and control through the composer's incorporation into the system.

The application of systems to art has evolved rapidly beyond these initial foundations and has become unavoidably present in most contemporary work due to technology. The nature of systematic conception often becomes an afterthought, avoiding the fundamental questions present in the creation of system aesthetics and focusing on facilitating the most efficient utilities. By working with CRBMs and FCRBMs that rely on a composer's choices and its own synthesis, I return to these questions. By focusing on the interactions of composers with the algorithms in this context, I've developed a generalized approach to creativity can be extended to composers working with timbral music composition systems.

### 3.1.2 Jevbratt's Interspecies Collaboration

With the advancement of these complex human-machine interaction via systems, the role of the composer comes to the forefront. Wherein previously the composer controlled creation through their aesthetic choices, the ceding of more creative elements to technology shifts the human-centricity of art to one of seeking balance within the system. With this shifting balance between composer and technological agents, new collaborative methods must be developed in order to deepen our understanding of such human-machine interactions.

Lisa Jevbratt's work with *Interspecies Collaboration*[32] provides many insightful guides to this end that parallel concepts present in my own research. In the introduction to her course, she states the purpose and benefit of working with non-human agents, opening ourselves to "learn things about our world we (quite literally) cannot imagine" and "to acknowledge their agency[...]to be our intellectual, emotional and spiritual partners in a quest for a sustainable environment for all of us to thrive within [32]."

The parallels with human-computer collaboration, especially present in my work with CRBMs and FCRBMs, are clear. By allowing algorithms to take a larger share in the creative process, by acknowledging their 'agency' in the artistic process, a more complex understanding of our artistic systems is assumed, allowing for material creation beyond the isolated vision of a composer. Expanding into this algorithmic agency empowers the human composer, providing investigative means into a completely different creative paradigm. Spawned from that paradigm are new ideas, a more developed understanding of systematic interactions, and a rich complexity for aesthetic growth. My timbral music composition systems generate material that is directed by the user, but defined and fully realized through the relationships within the system's agents, composer and algorithms.

Beyond the conceptual motivation, Jevbratt provides four useful forms which are used to develop agency in non-humans and are rich sources to draw from for the human-machine interactions of my compositional systems: *protocol*, a formalized rule system that defines the interactions between agents within a system to generate an output; *interference pattern*, the co-existence of two environments (whether they be physical, emotional, or semiotic), that creates a new environment as a result of their interference; *communication*, the development of reactions and responses between agents, listening, observing, and absorbing input and figuring how to respond accordingly, transferring information in response to the information received; and *limbic resonance*, "a symphony of mutual exchange and internal adaptation whereby two mammals become attuned to each other's inner states [33]."

Jevbratt's schema for non-human interaction provides a useful outline for interactions with algorithms creatively. While her focus resides in the living, species domain, there are clear parallels with human-computer interaction that are present in my research, especially from a collaborative perspective. The utility of my systems are creative in nature rather than one of straightforward functionality, opening composers to a completely different and potentially rich paradigm to explore aesthetics.

## 3.2 Musical Composers

While the basis for this research begins with system aesthetics, its closest conceptual relatives are found in musical composition, with composers addressing musical synthesis and control problems from a formal (Xenakis), timbral (Saariaho), and systematic (Bischoff and Tudor) perspective. The work of these composers displays the potential of integrating technological tools into aesthetic development, striving to advance the capabilities of human-machine interactions in concept and implementation. This potential can be uniquely realized using CRBMs and FCRBMs, expanding how these composers addressed their respective compositional challenges.

### 3.2.1 Systematic Form Creation: Iannis Xenakis

Iannis Xenakis use of stochastic processes and probability in composition lend itself directly to systematic music composition and developing technology to address the resulting conceptual issues. His novel approaches to incorporating mathematics, linear programming, computational methodology, and symbolic formalization of music composition blend seamlessly with the organizational and artistic framework laid forth by system aesthetics. His work confronts many of the same structural and theoretical problems being addressed by artists working in systems (see 3.1, defining his process in terms of music as ”an organization of [...] elementary operations and relations between sonic entities or between functions of sonic entities [34].” CRBMs and FCRBMs can be viewed from this lens, developing a series of algorithmic operations with relation to the data.

His approach is articulated in *Formalized Music* [34]. Xenakis defines his process of composition in eight fundamental phases:

1. Initial conceptions (intuitions, provisional or definitive data)

2. Definition of the sonic entities and of their symbolism communicable with the limits of possible means (sounds of musical instruments, electronic sounds, noises, sets of ordered sonic elements, granular or continuous formations, etc.)

3. Definition of the transformations which these sonic entities must undergo in the course of the composition (macrocomposition: general choice of logical framework, i.e., of the elementary algebraic operations and the setting up of relations between entities, set, and their symbols as defined in 2.); and the arrangement of these operations in lexicographic time with the aid of succession and simultaneity)

4.Microcomposition (choice and detailed fixing of the functional or stochastic relations of the elements of 2.), i.e. algebra outside-time, and algebra in-time

5. Sequential programming of 3. and 4. (the schema and pattern of the work in its entirety)

6 Implementation of calculations, verifications, feedbacks, and definitive modifications of the sequential program

7. Final symbolic result of the programming (setting out the music on paper in traditional notation, numerical expressions, graphs, or other means of solfeggio)

8. Sonic realization of the program (direct orchestral performance, manipulations of the type of electromagnetic music, computerized construction of the sonic entities and their transformations) [34].

Using this process as a scaffold for human-machine collaboration, the balance of agency in my timbral composition systems can be defined. With the exception of the first "initial conceptions" and final "sonic realizations," each of the phases can be achieved in my compositional systems by the design of the composer, a collaborative effort between composer and algorithm, or completely automated by the algorithm. In these phases, a weighted balance between each agent seems most conducive to facilitating creativity. For example, in the "definition of the transformations," composers provide higher level direction that could be experimented with, selecting the initiation data for the CRBMs (see 4.2) and defining external labels for the FCRBM (see 4). The results of this experimentation change the initial conceptions of the transformation, creating a human-machine feedback loop, where the composer adjusts to the synthesis of the algorithms. This dialogue and resulting development only occur in a system emphasizing interactivity and synthesis of undefined material. My music composition systems provide that due to their reliance on CRBMs and FCRBMs to autonomously generate timbral textures with respect to the choices of the composer (see 5.1 and 5.2).

Coupled with this process, Xenakis noted the advantages presented by computers to the compositional process, including the massive increase in processing capabilities, the ability to operate freely from a higher level of empowerment on musical material (i.e. form, input data), the ability to share this musical form directly through the vernacular of programming, and, through this dissemination, the ability for external composers, machines, and performers to instill their own 'personality' in their use of the compositional material.

He began realizing these ideas in the work *ST/10-1, 080262* (1962), a piece of stochastic instrumental music based on the scheme he designed for the earlier work *Achorripsis* (1957) and programmed for the IBM-7090. In the scheme, Xenakis defined a series of limits and rules for sonic sequences, realizing them using a probabilistic approach to form, specifically following Poisson's Law. He argued that even though elements of the piece appear aleatoric at first hearing, successive exposures to the piece form a rules-network that the listener begins to hear, organize, and innately associate with that composer's version of the piece. This method turns the composer into "a sort of pilot," defining and supervising the controls of a "cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream, constituting a new musical form [34]." After the initial composition of *ST/10-1, 080262*, other works were written in a similar form including *ST/48-1, 240162* for large orchestra, *Atrees* for ten soloists, and *Morisma-Amorisima* for four soloists.

This directed piloting of material through probabilistic distributions draws many conceptual parallels to the use of label driven factors in FCRBMs. Much like Xenakis' defined limits and probabilistic rules for sonic consequence, my composition systems allow composers to direct the timbral output to different textural centers and dynamic spectral events that are generated by the algorithm (see 4.4.2, 4.4.3, 5). Arriving at these textural centers is defined through the transitory capabilities of FCRBMs, continuously transforming timbre using an overlapping, autoregressive timeframe, presenting a machine generated, label induced shift of sounds. These textural transitions can be extrapolated to larger, formal distinctions, through the use of multiple layers of the compositional systems (as in 5.2) or in future work with deeper networks (see 6.4).

### 3.2.2 Timbral Synthesis and Manipulation: Kaija Saariaho

Much of the compositional work of spectralist Kaija Saariaho deals directly with manipulating timbre and sound processing. This navigation of the dynamic nature of audio transformations necessitates the development of tools that allow for the timbral manipulation of soundscapes.

From this basis, she incorporates digital sound processing and synthesis as a natural extension of her exploration of timbre. In *Lohn*, a work for soprano and electronics, she uses several transformation programs developed at IRCAM to realize the composition including *Chant programme* (resonance), *AudioSculpt* (cross-sythesis, phase-vocoder time stretching), and *Spatialisateur programme* [35]. In *Io*, an electroacoustic work for large ensemble and electronics, she accompanies and manipulates the ensemble's sounds in real time, digitally constructing soundscapes as a result of live processing with electronics and extending timbres in combination with pre-recorded sounds [36].

In Six Japanese Gardens, Saariaho even "voluntarily reduced" the percussionist instrumental pallet, only for "the reduced colours [to be] extended with the addition of an electronics part." The specific dimensionality reduction (similar to the reduction by abstracting data to the hidden layer as explained in 6.2.1) of the performers responsibility for timbre "give[s] space for the perception of rhythmic evolutions," allowing for a deeper conceptual expression of the composer by the performer [37]. Saariaho shapes her composition to the unique advantages presented by a percussionist (i.e. perception of rhythmic development) and the electronics (i.e. timbral expansion and transformation), creating an effective expression of her intent.

In each of her operatic works *L'Amour de loin* (2000), *Adriana Mater* (2005), *Emilie* (2008), and *Only the Sound Remains* (2015), Saariaho designates specific timbral transformations of the vocalist, spatializing, amplifying, and blending acoustic spectrum with digital, creating fused soundscapes that are a direct result of technological capabilities. Conceptually, this fusion can represent symbolic narrative elements (i.e. processing the voice to reflect who the vocalist is thinking about in *Emilie*), augmented electro-acoustic character (i.e. merging vocal soloist and with subtle electronic timbres in *L'Amour de loin* and *Adriana Mater*) or practical expansions in timbre (i.e. electronic amplification in all of the operas, repeated playback of audio samples or elongating acoustic sounds in *Only the Sound Remains*), resulting from the interactions of the composite parts.

The composition of sonic consequences as the realization of the composer, performer, and technological utility is the precise dynamic I wish to accomplish in my composition systems. This engagement of the agents of a system to facilitate aesthetic goals is a direct extension of the systematic applications developed by system aesthetics (see 3.1). Extending this engagement to one of dynamic interaction (i.e. composer and algorithm relying on each other to compose), rather than static facilitation (i.e. composer implementing direct audio processing) is how my work evolves from Saariaho's.

*Prisma* is a particularly illuminating creation by Saariaho, coupling a more standard audio album of her work with interactive composition software. The software provides the user with a collection of multimedia data, including original texts, sounds, and videos, and invites them to "clarify the relation between musical notation, the gesture of the performer, and the musical result [38]."

While mainly serving the purpose to educate the listener on contemporary composition techniques and promote a better understanding of her own work, *Prisma* also acts as a curated, compositional system displaying the capabilities of aligning creative conception with technology. This system invites users to compose freely in a variety of graphical representations (i.e. solfege syllables, spectral depictions of timbre, amplitude variations), connecting their understanding to interactions created by Saariaho. In this instance, unlike her composed works mentioned above, Saariaho created a collaborative intermediary with the user through technology, balancing all three agents to create a shared conceptual experience. This collaborative realization most closely connects with my own implementation of CRBMs and FCRBMs in artistic application (see 5.2).

### 3.2.3 Systematic Music Composition: John Bischoff and David Tudor

John Bischoff's current compositional work and former work with The Hub [39] establishes a relationship with systems as part of the creative process and performance, integrating the composer-performer directly into an interactive computer network. The Hub composed through software and network design, programming compositions through a set of interaction schemas resulting in improvised performance. John Bischoff describes gives an example in a description of his piece *Perry Mason in East Germany*:

> Each of the six players runs a program of his own design which constitutes a self sustaining musical process. Each program is configured so that it can send three changing variables important to its operation out to the Hub and also to receive three variables from other players. Each player reads the variable put out by three different performers, and sends out for use by three different performers as well. This relationship of mutual influence results in a network structures that often yields a special kind of musical coherence. [40]

As their work extended, the members of The Hub began to exchange the code of their electronic instruments, using computers directly in the synthesis process. Complicated, non-linear relationships between the multiple performers and the development of those relationships through computer instruments created a complex artistic performance system that my work draws from. Specifically, the ability to factorize the FCRBM, learning the connections between the composite elements of the algorithms and defining the protocol with which these elements interact draws directly from the performative structures of Bischoff (see 5). The development of my composition systems have established of several algorithmic instruments that are dependent on each other for musical synthesis. This has led to factoring beyond the individual algorithmic instruments, into an fully connected network where the relationships of the ensemble are modeled and able to be composed (see 5.2). These higher level structures draw directly from the theory and work of the Hub.

Bischoff's album *The Glass Hand* specifically works with transforming timbres drawn randomly from MIDI synthesizers, creating sonic consequences "built around sonic properties discovered in these MIDI devices and, as such is derived from the electronic system itself [41]." This album more directly drives my current work, providing evidence of systematic timbre transformations that use the internal dynamics of that synthesis to define the aural outcomes. These ideas are realized in the manipulation of FCRBMs through external labeling, providing the machine synthesis with an overarching intent, but allowing the machine to map the timbral qualities of the sounds and their transformations largely independent of the composer (see 4.4.3).

Additionally, the analog-electronic networks and systems of David Tudor [42] develop and shift through their internal connections, characteristics, and the resulting consequences of those connections. The compositional focus on the development of the system provides an invaluable example of successfully addressing the complex nature of multi-layer, non-linear performance systems in expressing aesthetic concerns. This is especially apparent in Tudor's *Neural Synthesis* [43].

In an attempt to develop a computer system emulating Tudor's analog electronic performance system, Forrest Warthman and Tudor began developing an approach to digital/analog synthesis in live performance. During this work, they were introduced to a neural-network microchip that consisted of "64 non-linear amplifiers with 10240 programmable connections [44]" that could be interconnected with varying connection strength, mimicking neural links. With the chip generating and routing signals across the performance network, Tudor was able to interact and respond based on his own aesthetic choices, quickly generating complex sonic atmospheres in performance. These atmospheres were only realized through interacting with the vast electronic network of which he directed toward desired sonic textures. Similar to Xenakis' compositional piloting (see 3.2.1), Tudor provided himself as the higher level control structure for the performance network, similarly finding parallels with the FCRBMs' user-labeled guidance (see 2.2.4).

As a composer, Tudor drew upon resources that are both flexible and complex but ultimately reliant upon his ability to direct and manage the network. He used custom-built modular electronic devices and his compositional method employed musical as well as design and manufacturing strategies. The choices of specific electronic components and their interconnections defined each piece in both composition and performance. The music unfolds through large gestures in time and space, macro-directed by the composer-performer and micro-realized by the devices.

Ultimately, the very control and design that Tudor attempts is very similar to the type of direction I attempt with CRBMs and FCRBMs in my composition systems (see 5.1 and see 5.2). I define the sonic textures through the construction of audio material from which the algorithms define their interactions. After the CRBMs learn this material and the connecting patterns that will transform it accordingly, the directed manipulation of the textures is reliant on the composer with either the external labeling of a FCRBM or internal tuning of the algorithmic parameters, just as Tudor depended on his personally constructed circuitry design and live performance interactions with his networks. The interplay between the composer and the network/algorithms cannot be divorced from the resulting sonic consequences, realizing composition through design and human-machine collaboration.

## 3.3 Summary

In this chapter, I provided the art theory and concepts found in system aesthetics and music composition that motivate the creation of my timbral music composition systems, articulating interdisciplinary methods of addressing increasingly complex technological and artistic problems. Through the citation of specific examples within these artistic domains and connecting them to the unique capabilities of CRBMs and FCRBMs, I give a precedent for my research and the creation of timbral synthesis and control systems, showing that the development of these creative systems is integrated with technology.

# CHAPTER 4

# MODEL TESTS AND VALIDATION

In this chapter, I developed CRBMs and FCRBMs to perform timbral synthesis and control tasks based on audio data, evaluating the algorithms' performance capabilities in a variety of experiments. In section 4.1, I describe the general data preparation and testing process for the experiments. I then describe three specific aspects of timbral synthesis I tested: modeling unique instruments without segmentation in section 4.2, modeling different pitches played by singular instruments in section 4.3, and creating compositional utilities in section 4.4. These fundamental experiments algorithmically realize compositional possibilities and create an efficient, systematic approach to timbral synthesis and control.

## 4.1 General Testing Process

For testing and synthesis in these experiments, I trained CRBMs and FCRBMs with spectral data generated from real audio. Using this audio training data, I explored the internal parameters of the algorithms, seeking the optimal set that would deliver the highest accuracies in the experiments while maintaining computational efficiency.

### 4.1.1 Data Representation of Audio

For each experiment, I constructed a model to synthesize target classes, training that model using spectral representations of sample audio. To create the spectral representations, I used a short-time Fourier transform (STFT) [45] on the audio. With this STFT, I was able to adjust the number of samples over which the STFT was computed (STFT window size), how frequently the STFT was performed (hop), how the STFT windows overlapped (i.e STFT window size = 4096 with a hop = 4096, no overlap; STFT window size = 4096 with a hop = 2048, 50% overlap), and what type of tapering window (if any) was applied to the STFT samples. This spectral data was then normalized using the first bin produced in the STFT of all the audio samples and standardized with the mean and standard deviation of the spectral data.

While I did not go deeply into testing ideal STFT parameters (i.e. STFT window size, hop, overlap, tapering window), I did run all experiments with two different STFT parameter settings (4096 frames with a 2048 hop and a Hanning tapering window; 512 frames with a 256 hop and a Hanning tapering window), resulting in very similar outcomes. In general, the larger STFT window (4096/2048) resulted in higher average accuracies and more efficient computation. Additionally, the 4096/2048 STFT has been used in other CRBM applications that deal with audio and function similarly to my research [46] [47]. Considering these factors, I report the result of the experiments using a 4096/2048 STFT with a Hanning tapering window, unless otherwise noted.

### 4.1.2 Internal Model Parameters

In comparing the performance of the different models, I tested two different algorithms for the experiments: a single layer CRBM and a single layer FCRBM. Each model had an adjustable number of hidden units, model order, and number of contrastive divergence steps to be taken in learning. The FCRBM had an adjustable number of factors and features to be used in learning, attaching composer-defined labels to the different training classes.

While Hinton presents several empirically validated starting points for setting the internal model parameters of CRBMs [48], a more thorough investigation, especially with regards to the audio domain and FCRBMs, would be beneficial to my research. I extensively tested the internal model parameters (i.e. number of hidden units, number of features, number of factors) and iteration numbers (i.e. steps of contrastive divergence, steps of Gibbs sampling) of the algorithms, seeking the optimal set for accurate synthesis in each experiment.

I divided the internal model parameters into three separate groups: *dimensionality reduction/dimRed*, setting the value of the internal parameters to one fourth the number of features in the data; *constant*, setting the number of parameters to the same as the number of features in the data; and *augmented*, setting the value of the parameters to twice the number of features in the data. For a 4096/2048 STFT, this resulted in dimRed = 512 parameters, constant = 2049 parameters, and augmented = 4098 parameters.

For the number of iterations, I tested in three groups: one iteration (i.e. CD and Gibbs steps = 1); five iterations, and twenty iterations. While CD and Gibbs steps did not have to coincide, I chose these three groups as indicators for further investigation dependent on the test results (for complete list of model conditions tested, see B).

Learning was optimized by parallelizing and running the algorithm on the graphic processing unit (GPU). Additionally, stochastic gradient descent in the algorithms was optimized with ADAGRAD [49]. For details, technical specifications, and justification for use, see A. For these experiments, training was stopped after 3000 epochs in order to compare the performance across the different algorithms and to determine the ideal parameter configuration for each condition.

### 4.1.3 Model Synthesis and Evaluation

Once each model was trained, it synthesized the timbres it modeled (see experiments 4.2 and 4.3) or created new, machine-synthesized timbres (see experiment 4.4). While each test worked with unique initialization data relative to the task, several properties were shared across tests. The only input from the composer included a number spectral frames equal to the order of the model (i.e. order = 5, number of sample frames needed = 5) to initialize the algorithm, the number of frames to be synthesized, and, for the FCRBM, a label for each corresponding frame. The resulting output was then completely synthesized by the model without intervention. Initialization data was made into a spectral representation using an STFT with the same parameters used in training. After the data had been synthesized, the spectral output of the model was then run through an inverse fast-Fourier transform (IFFT) with overlap and add, reverse data standardization and normalization, and amplitude compression in order to create audio.

After generating the audio, the synthesis was classified using two separate models: multi-class binary support vector machines using one-versus-one error-correcting output codes (ECOC(SVM)) and a multi-class naive Bayes model (NB).

Error-correcting output codes take multi-class (more than two classes) classification and turns it into a number of binary classification tasks, in this case using binary SVM classifiers. Given the number of unique classes (k), the number of binary models used is equal to k(k-1)/2. I specifically used Matlab's *fitceococ* function to perform this classification. I chose this model as it had been demonstrated to be a reliable and efficient classification method for handling highly dimensional data in comparison to other multi-class approaches [50] [51] [52].

The multi-class NB model provided an additional method of classification that provided a baseline for synthesis accuracy. I specifically used Matlab's *fitcnb* funcition to perform this classification. This model serves as a fundamental standard in statistical learning and has been frequently used for as a baseline for comparing classification accuracies [53] [54].

The classifiers were trained with the complex magnitude of the STFT of real audio samples, reflecting the different tests respective classes. For most tests, I used at least 10 times the amount of training data as test/synthesized data, but specifics varied according to tests. In order to evaluate a ground truth for both of these models, I tested the trained models with real samples, resulting in the accuracies reported in 4.1.

Table 4.1: Accuracies resulting from ground truth test on ECOC (SVM) and NB models

| **Strings** | **Winds** | **Percussion** | **Violin Scale** | **Oboe Scale** | **Bells Scale** |
|---|---|---|---|---|---|
| .97/.81 | 1.00/.82 | .86/.80 | 1.00/.97 | 1.00/.95 | 1.00/.93 |

The results of the ground truth tests show the superiority of the ECOC (SVM) to the NB models in accurately classifying audio spectrum. In qualifying the classification results of the synthesis experiments, I anticipated that sounds classified by the ECOC (SVM) would result in higher accuracies in comparison to the NB models.

### 4.1.4 CRBM Training Methodology

While FCRBMs is designed to model multiple classes directed by user-defined labels, CRBMs rely on initialization data to orient synthesis. This led me to initially test two separate implementations of CRBMs to model multiple classes of data: a single model of all classes combined, relying on different initialization data to synthesize the different audio classes ($CRBM_{all}$) and multiple, separate CRBMs trained for each unique class ($CRBM_{sep}$). While the performance of the $CRBM_{sep}$ had a higher accuracy more inline with expectations of the model, being able to synthesize a variety of sounds related to the classes, the $CRBM_{all}$ model tended to drive all synthesis toward a mix of all the sounds, generating very similar timbres regardless of initialization. I believe this was mainly due to the lack of learning done in the set/fixed number of iterations. Given a higher number of epochs to train, the $CRBM_{all}$ models performance improved. From these initial tests, I decided to only report the accuracies resulting from $CRBM_{sep}$, as it was able to achieve desired tasks more accurately and in less time using a comparable number of iterations in training.

The overall accuracy for the classification tests was measured as the correctly classified synthesis frames divided by the total number of frames classified. Thus, if a synthesized sample of 100 frames of 'CLASS A' was classified as 'CLASS A' in 90 of the frames, the classification would be reported as 90% accurate.

## 4.2 Modeling Unique Instruments Without Segmentation

For the first synthesis experiment, I attempted to synthesize unique instruments without segmentation from the same musical families in three separate groups: strings, woodwinds, and pitched percussion. For each of the families I tested four different instruments playing the same pitch so as to emphasize the unique timbral qualities of the instruments.

### 4.2.1 Test Construction

For the parameters in this test, I set the STFT window size to 4096, the hop to 2048 (i.e. 50 percent overlap), and applied a Hanning tapering window. This resulted in a total of 2049 spectral features for each observation. From the resulting STFT, the internal parameter settings were set to dimRed = 512 (r), constant = 2049 (c), augmented = 4098 (a). The order of the model was set to 4 frames. Once trained, each model synthesized 50 frames of the specified class, being initialized with data from that specified class.

For the string family, I synthesized the violin, viola, cello, and bass, all playing the pitch A3. For classification, I created an ECOC(SVM) model using 6 (k=4) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 703 STFT samples, evenly distributed across the classes.

For the woodwind family, I synthesized the oboe, bassoon, clarinet, and flute, all playing the pitch B4. For classification, I created an ECOC(SVM) model using 6 (k=4) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 1096 STFT samples, evenly distributed across the classes.

For the pitched percussion family, I synthesized the xylophone (rosewood mallet), marimba (yarn mallet), crotale (brass mallet), and bells (brass mallet), all playing the pitch A4. For classification, I created an ECOC(SVM) model using 6 (k=4) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 790 FFT samples, evenly distributed across the classes.

### 4.2.2 Test Results

For complete results of the various conditions tested, see B.2. The tests resulted in models performing with the highest accuracies given the conditions in B.5.

Table 4.2: Highest Resulting Models from Experiment 4.2 with Accuracies from ECOC(SVM)/NB in comparison to ground truth (GT) accuracies. Models are CRBM (C) or FCRBM (F) with reduced (r), constant (c), or augmented (a) features sets using 1, 5, or 20 Gibbs steps.

| **Strings** | Accuracy | **Winds** | Accuracy | **Percussion** | Accuracy |
|---|---|---|---|---|---|
| $C_{r1}$ | .95/.96 | $C_{r20}$ | .96/.79 | $C_{r5}$ | .96/.98 |
| $F_{c1}$ | .91/.85 | $F_{c5}$ | .86/.68 | $F_{c20}$ | .63/.67 |
| GT | .97/.81 | GT | 1.00/.82 | GT | .86/.80 |

### 4.2.3 Test Discussion

When looking at the criteria tested, a variety of outcomes resulted. In comparison to the ground truth accuracies, the CRBMs were able to perform nearly as well as the ground truth tests, even outperforming them in the percussion case. FCRBMs consistently underperformed the ground truth and CRBM accuracies. From these results, it appears that CRBMs would be ideal for modeling more subtle differences of timbre as found in similar instruments (i.e. instruments from the same family) playing the same pitch. This could be due to the exclusive nature of the CRBM models, training one model for each class, providing a very focused algorithm that did not have to generalize beyond a narrow set of sounds (i.e. each of the CRBMs was only trained with the targeted data class, not needing to model beyond that specific timbre).

FCRBM synthesis was most often incorrectly classified in the decay/silent portions of the sound. In these spaces, the FCRBM synthesized what sounded like an average of all the sounds it was modeling at a diminished amplitude. In the training samples, the silence following the played pitch would sound the most similar in comparison to the rest of the sound, thus be much more difficult to distinguish in synthesis. This potentially could be resolved by modeling silence as an additional class to the instruments, providing the FCRBMs with a more precise reference to model. Classification results could also be improved by segmenting training samples at a specific amplitude threshold and training the algorithms with sounds that did not have silence. While this may improve classification results, the resulting sound itself would most likely sound less like the target classes, distorting the amplitude envelope beyond the unique characteristics of the instruments. For example, pitched percussion produces a very specific amplitude envelope that very rapidly decays. If the training samples were thresholded to sounds that exceeded a specific amplitude, the samples would not reflect the percussive nature of the instrument, eliminating the characteristic rapid decay (see 4.4.1 for an exploration of compositional envelope manipulation).

In this decay, return to silence portion of the synthesized sounds, models would occasionally generate timbral material independent of the training samples. This is especially evident in the pitched percussion, where a rapid decay immediately follows the attack. This resulted in synthesis that produced a rapid succession of attacks as heard in the xylophone and marimba samples in the $F_{c20}$ synthesis. While the sonic quality of the sound can be heard as instrumentally distinct, the dissimilarity of the amplitude envelope and the resulting difference in the sounds decay led to misclassified frames, classifying those synthesized sounds with longer decay tails as ‘bells,’ as seen in the confusion matrix for $F_{c20}$ (see 4.3).

Table 4.3: Resulting confusion matrix using ECOC(SVM) classification of percussion timbres using the $F_{c20}$ algorithm

| $\mathbf{F_{c20}}$ | Xylophone | Marimba | Crotale | Bells |
|---|---|---|---|---|
| Xylophone | 1 | 1 | 1 | 0 |
| Marimba | 2 | 8 | 0 | 0 |
| Crotale | 0 | 0 | 46 | 0 |
| Bells | 47 | 41 | 3 | 50 |

When given an augmented number of features to model the data, accuracy dropped. Given this outcome and that each model was trained for a specific number of epochs, it is clear that models with higher complexity required more iterations to train more accurate models. For example, when a CRBM with a string dataset using an augmented number of features with 1 step contrastive divergence and Gibbs sampling ($C_{a1}$) was trained for 3000 epochs, its synthesis achieved 25.5/29% accuracy in ECOC(SVM)/NB classification tests where the majority of the synthesis was classified as a singular class (see 4.4). When the test was repeated for 30,000 epochs, ten times the iterations of the initial test, the $C_{a1}$ model's synthesis achieved an accuracy of 63.5/64.5%, with a much more evenly distributed classification than the initial test (see 4.4).

Table 4.4: Resulting confusion matrix using ECOC (SVM) classification of violin timbres using the $C_{a1}$ algorithm for 3000 and 30,000 epochs

| $\mathbf{C_{a1}3000}$ | Violin | Viola | Cello | Bass |
|---|---|---|---|---|
| Violin | 7 | 10 | 0 | 10 |
| Viola | 0 | 3 | 9 | 0 |
| Cello | 43 | 37 | 41 | 40 |
| Bass | 0 | 0 | 0 | 0 |

| $\mathbf{C_{a1}30000}$ | Violin | Viola | Cello | Bass |
|---|---|---|---|---|
| Violin | 32 | 19 | 2 | 4 |
| Viola | 4 | 27 | 11 | 6 |
| Cello | 8 | 4 | 37 | 9 |
| Bass | 6 | 0 | 0 | 31 |

This level of accuracy was achieved by the a FCRBM for the same parameters ($F_{a1}$) in only 3000 epochs (65/66% accuracy, see B.2). This discrepancy in performance occurs with multiple CRBMs that used the augmented parameter set ($C_{a1} = .23$, $C_{a5} = .20$, and $C_{a20} = .22$). This suggests that CRBMs perform better given specific parameter conditions while FCRBMs generalize better as the complexity of the model increases. This is seen in the variance of the resulting accuracies, with CRBMs having a wider range. While the average accuracies of the algorithms are relatively close (CRBM = .709 to FCRBM = .683 in ECOC (SVM) classification, see B.5), the variance of CRBMs was much higher than the variance of FCRBMs (CRBM variance = .05, see B.7 and FCRBM variance = .01, see B.4).

Where certain, more complex CRBMs needed more iterations to train, less complex models performed highly accurate synthesis. For example, CRBMs with the string family dataset using an augmented number of features with 20 steps of contrastive divergence and Gibbs sampling ($C_{a20}$) achieved a 24% accuracy in synthesis while a FCRBM modeling a dataset using a constant number of features with 1 contrastive divergence and Gibbs step ($F_{c1}$) achieved 91% accuracy in synthesis, using the same amount of training. Taking it a step further, a CRBM modeling the string family dataset with a reduced number of features with 1 contrastive divergence and Gibbs step ($C_{r1}$) was able to achieve 95% accuracy in synthesis (see B.6).

From a synthesis perspective, this shows that if the goal of the task is to model a narrow range of specific timbres efficiently, one can use the simplest model and achieve very high accuracies. If the task includes being able to navigate between classes within the same model (arguably a more efficient approach, as only one FCRBM would be needed to be trained as opposed to 4 CRBMs), this also can be achieved at comparatively low levels of parametric complexity. The versatility in a singular multiple class model (i.e. FCRBMs) is more appealing from a compositional perspective than using multiple single class models (i.e. CRBMs) in that it allows for synthesizing transitions between classes, different temporal-timbre blending approaches, and a more continuous way to create different sounds (see 4.4.3).

In all cases for the FCRBMs, going from a reduced set to a constant set of features achieved the same or improved accuracy in synthesis, on average increasing the accuracy significantly (see B.4). Inversely, in every CRBM test, going from a reduced set to a constant set of features reduced the accuracy of synthesis (see B.7). This suggests that if the goal of the task is to reduce the dimensions of timbre space, it can be achieved on a sound to sound basis, but in order to obtain more accurate models for multi-class synthesis, a comparable number of features to the presented dimensions of the training data are necessary.

Augmenting dimensions in CRBMs did not necessarily improve synthesis, at least at the tested number of training iterations (see B.7). This supports a need for more training to accurately model higher complexity, as demonstrated in previous results.

Beyond the highlighted issues with modeling sound decays and silence, the CRBM and FCRBM did comparatively well in their idealized model contexts, achieving accuracies that were very close to the ground truth tests. In looking at an aggregate of the results across each of the tests, confusion heat maps display a diagonal pattern, suggesting that the models were able to accurately synthesis the different timbres across instruments (see B.2). The clearest distinctions in training samples and synthesized sounds were found in the attack portion of the training samples, and these portions of synthesis were almost always classified accurately.

The results of this experiment provided concrete evidence that CRBMs and FCRBMs are capable of modeling subtle differences of timbre found within musical instrument families. By investigating the internal parameters of the algorithms, I was able to build upon previous, more generalized research [48] and find optimal parameter sets specific to modeling timbre in this context. This initial test also provided insight into how these algorithms model the amplitude envelope of sounds, provoking an exploration into the segmentation of synthesis (see 4.4.1). It also demonstrated what parts of the sounds these algorithms were ideal for modeling, instigating compositional forays into the synthesis of dynamic material with a sustained amplitude, reoccurring patterns, and continuous, incremental evolutions of timbre (see 4.4.2).

## 4.3 Modeling Different Pitches of Singular Instruments

For my second experiment, I synthesized different pitches of singular instruments without segmentation in three separate groups: a violin playing a G major scale, an oboe playing a D major scale, and bells (brass mallet) playing an A major scale. The scales were chosen in ranges that would limit instrumental register shifts (violin and the oboe) and/or be in the middle range of the instrument (oboe and bells).

### 4.3.1 Test Construction

For the parameters in this test, I set the STFT window size to 4096, the hop to 2048 (i.e. 50 percent overlap), and applied a Hanning tapering window. As in the previous test, this resulted in a total of 2049 spectral features for each observation. From this, the internal parameter settings were set to dimRed = 512 (r), constant = 2049 (c), augmented = 4098 (a). The order of the model was set to 4 frames. Once trained, each model synthesized 50 frames of the specified class, being initialized with data from that specified class.

For the G major scale played by a violin, I synthesized the following pitches: G3, A3, B3, C4, D4, E4, F#4, and G4 . For classification, I created an ECOC(SVM) model using 28 (k=8) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 1524 STFT samples, evenly distributed across the classes.

For the D major scale played by an oboe, I synthesized the following pitches: D4, E4, F#4, G4, A4, B4, C#5, and D5. For classification, I created an ECOC(SVM) model using 28 (k=8) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 1128 STFT samples, evenly distributed across the classes.

For the A major scale played by the bells,, I synthesized the following pitches: A4, B4, C#5, D5, E5, F#5, G#5, and A5 . For classification, I created an ECOC(SVM) model using 28 (k=8) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 3048 STFT samples, evenly distributed across the classes.

### 4.3.2 Test Results

For complete results of the various conditions tested, see B.6. The tests resulted in models performing with the highest accuracies given the conditions in 4.5.

Table 4.5: Highest Resulting Models From Experiment 4.3 with Accuracies from ECOC/NB in comparison to ground truth (GT) accuracies. Models are CRBM (C) or FCRBM (F) with reduced (r), constant (c), or augmented (a) features sets using 1, 5, or 20 Gibbs steps. '*' indicates multiple models achieved the same accuracy.

| **Violin Scale** | Accuracy | **Oboe Scale** | Accuracy | **Bells Scale** | Accuracy |
|---|---|---|---|---|---|
| $C_{r1}$ | .98/.79 | $C_{r20}$ | .94/.97 | $C_{r1}$* | 1.00/1.00 |
| $F_{c5}$ | .97/.76 | $F_{c5}$ | .96/.92 | $F_{c20}$* | 1.00/.99 |
| GT | 1.00/.97 | GT | 1.00/.95 | GT | 1.00/.93 |

### 4.3.3 Test Discussion

In comparison to 4.2, resulting accuracies improved across the tests. Even though these tests involved more classes (8 versus 4), pitch as the discriminating feature was more distinct than the subtle instrumental differences within musical families. This was shown in the resulting CRBM and FCRBM accuracies as well as the high classification accuracy in the ground truth testing. Many similar results from the first experiment are paralleled in this experiment: more complex models needed more iterations to achieve higher levels of accuracy, leading to lower accuracies in the more complex CRBM models (i.e. the constant and augmented parameter sets); CRBMs achieved high accuracies with reduced parameter sets; FCRBMs improved performance when increasing the size of the internal model parameters from *reducedDim* to *constant*.

One especially noticeable difference between the experiments was the overall improved performance of the FCRBMs. FCRBMs performed significantly better than CRBMs on average (CRBM = .700 and the FCRBM = .873 using ECOC (SVM) classification, see B.9). This reinforces the idea above that FCRBMs generalize better than CRBMs in multiclass tests, where models become more complex and apply a more diverse set of training data. The significant jumps in accuracy when increasing the number of features from a *reducedDim* to a *constant* FCRBM (see B.8) complement this idea, and can be accounted for given the need for a higher dimensioned abstraction to effectively map more complex datasets.

As the number of CD/Gibbs steps increased, the average accuracy of the algorithms' synthesis improved (see B.9). These improvements in accuracy were more in line with expectations, that models would be able to synthesize more accurately if given more iterations to train and generate data. This is in contrast to the first experiment where the lowest accuracies were found after 5 iterations (see B.5).

Across the tests, the bells resulted in the highest accuracies, resulting in perfect classification in some cases. This could be a result of the consistency in the training data, with very little variation between training samples (i.e. there is little difference between percussive instruments beyond the attack, whereas violin and oboe can add vibrato or change timbral texture within the sustained components of the sound), reinforcing the central attribute that was being recognized, pitch. Wide vibratos and even variance in tuning across the train samples resulted in a more diverse training set for the violin and oboe, perhaps contributing to the less focused synthesis in comparison to the bells.

Most importantly, CRBMs and FCRBMs were able to model several different timbres with a higher accuracy when the modeled timbres were distinct. In the first experiment, efforts were taken to isolate more subtle timbral differences between instruments, keeping all sounds on the same pitch for each test resulting in much smaller differences in their spectral representations.

This led to samples that shared many similar characteristics, especially in the harmonic instruments of the string and woodwind families where the amplitude of the same harmonics significantly overlap.

In the second experiment, there were much clearer distinctions between the audio classes. Overlapping harmonics were not as similar (i.e. a G3 and G4 from the violin shared many of the same harmonics, but the different pitches had different amplitude distributions for those harmonics) and didn't occur as frequently as in the first experiment (i.e. in the first experiment, all classes had overlapping harmonics, in the second experiment, only select pitches overlapped). Using a training dataset with distinct timbral differences across the classes demonstrated the capability of these algorithms to synthesize a diverse range of timbres with high accuracies.

The results of the second experiment demonstrated another important timbral quality to be considered when using these algorithms for synthesis. While the synthesis from the first experiment resulted in higher accuracies when timbres were synthesized with sustained amplitudes, the synthesis from the second experiment resulted in higher accuracies when synthesizing clearly distinct timbres. Ideal sounds to synthesize and control compositionally using CRBMs and FCRBMs would have sustained amplitudes and be timbrally distinct, providing the algorithms with a clear aural pallet from which to compose its abstract representation. The second experiment confirmed trends in parameter selection found in the first experiment, giving a clearer structure to what types of parameters work best for modeling timbre.

Through the results of the first two experiments, I developed a better understanding of what type of timbral data CRBMs and FCRBMs model well and the corresponding parameter sets that provide the optimal synthesis of that timbre. Using these findings as a foundation for synthesizing and controlling musical timbre, I explored methods to leverage the unique capabilities of these algorithms in the creation several compositional utilities.

## 4.4 Compositional Utilities: Modeling Dynamic Envelopes of Singular Instruments and Sustained Dynamic Elements

For my final experiment, I created music composition utilities to synthesize and control music timbre, building on the methods and insights gathered from the first two experiments. Specifically, I manipulated synthesized sounds through the design of the algorithms, creating machine-driven compositional tools that allow composers to create sonic material and develop collaborative interactions with the machine agents of the systems.

I focused on three utilities targeting aspects of music composition that I felt would benefit from the unique capabilities of CRBMs and FCRBMs. I started by modeling the dynamic envelopes of sounds, synthesizing the attack, sustain, and release of singular instruments. I then synthesized distinct, sustained textures with consistent amplitudes, extending timbral synthesis within textures indefinitely. Finally, I designed a utility that synthesizes machine-driven timbres and timbral transitions from limited composer direction, fully realizing the capability of the algorithms to autonomously create timbres.

### 4.4.1 Dynamic Envelopes of Sounds

In order to explore the dynamic envelopes of sounds, I used samples from three real instruments (a crotale, a violin, and a tam-tam) playing a single pitch. I manually divided the instrumental samples into 3 different segments: the attack, the sustain, and the release.

For the parameters in this test, I set the STFT window size to 512, the hop to 256 (i.e. 50 percent overlap), and applied no tapering window. This resulted in a total of 257 spectral features for each observation. From this, the internal parameter settings were set to be constant (i.e. 257), the order of the model was set to 15 frames, and used 1 step of CD/Gibbs sampling.

The first task was to accurately synthesize the modeled sounds without manipulating the durations of the envelope segments, providing labeled guidance for the average number of frames for each envelope segment (i.e. if the average attack length of the training set was 25 frames, I synthesized 25 frames labeled as ‘attack’). The second task was to experiment with the lengths of the different segments, providing labeled guidance that was much greater than the average length of segments (i.e. if the average attack length of the training set was 25 frames, I synthesized 50-100 frames labeled as ‘attack’), hearing what effect this would have on the synthesized material.

For classification, I reconstructed sounds using the envelope segments and classified them as complete samples of the instruments (i.e. crotale v tam-tam v violin) as opposed to individual envelope segments. I chose to evaluate the utility this way as it was more in line with the expectation of synthesis (i.e. creating instrumental sounds that could be digitally manipulated via their envelopes) and less susceptible to the manual marking of envelope segments. I created an ECOC(SVM) model using 3 (k=3) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 5733 STFT samples, evenly distributed across the classes.

In the first task, reconstructed sounds were successfully synthesized for each instrument, being recognized with 100% accuracy as a crotale, violin, and tam-tam respectively, without perceptible differences between real and synthesized samples.

In the second task, I experimented with synthesizing different lengths of the various envelope segments, attempting to synthesize sounds that possessed similar timbral characteristics but were able to be extended in time. I found the amplitude within the envelope segments of the training data greatly affected the extent with which the synthesized audio could be accurately recognized as the instrument it modeled, most evident in the sustain segment of the envelope.

For example, the sustain of the crotale was able to be extended to four times its average length due to the model learning a relatively even amplitude from the training samples, with very little variance. This allowed the model to synthesize a continuous, extended sustain segment, with a smooth transition to the decay segment of the sound.

Unlike the crotale, the violin was only able to be extended to twice its average length, due to the model learning a slight crescendo during the sustain segment of the training samples. When extending beyond this length, the model augmented the crescendo to the point of distortion, effectively created a completely different sound from the training data, unrecognizable as a violin.

The tam-tam was also only able to be extended to twice its average length, but in contrast to the violin, this was due to the model learning a slight decrescendo during the sustain segment. Unlike the violin which increased amplitude to eventual distortion, the tam-tam sustain segment diminished to an inaudible level, essentially incorporating decay elements into the extended synthesis.

In classifying theses extended sustain sounds (a 4x crotale, a 2x violin, a 2x tam-tam), the models were able synthesize samples that were classified with a 98.9% accuracy (see 4.6).

Table 4.6: Resulting confusion matrix using ECOC (SVM) classification for extended synthesis of the sustain portion of the sounds based on dynamic envelope

| **ASR Envelope** | Crotale | Tam-Tam | Violin |
|---|---|---|---|
| Crotale | 959 | 4 | 0 |
| Tam-Tam | 0 | 4669 | 31 |
| Violin | 31 | 1 | 519 |

These learned developments were also present in the much less consistent amplitudes of the attack and decay segments. Extended attack segments quickly distorted beyond recognition whereas decay segments faded to silence and remained there for the duration of the synthesized sample. While the FCRBM was not able to synthesize extended attacks without distortion or decays that remained audible from the training set, the sustain segment of the envelope could be extended indefinitely, effectively creating a dynamic synthesizer capable of being played across timbres.

The versatility of the FCRBM to adapt to a variety of envelope mappings in different ways could be accounted for in adjusting the parameters of the model itself and analyzing the defining characteristics of the sampled instruments' original envelopes. With percussive instruments, where there is an almost instantaneous decay after the attack, careful consideration has to be given to what kind of sustain segment the model would be synthesizing. Both the crotale and the tam-tam models were able to synthesize sustain sounds very similar to the instruments they were modeling, but in doing so, took away the characteristic 'percussive' envelope, effectively generating a different type of instrument.

The violin, with a much more well defined sustain segment that can inherently be extended indefinitely, took on the characteristics of the training samples (i.e. a crescendo) creating a stylistic affect when extending synthesis. A potential future direction for expanding these types of models would be to train a variety of sustain styles (e.g. crescendo, even, decrescendo) that could then be used to define the characteristics of the synthesis.

The exploration of controlling and synthesizing the envelopes of instrumental timbres provided a more developed perspective of how sustained textures should be synthesized from a compositional perspective. The new evidence from this utility, coupled with the results from the first two experiments, provided a path to synthesize sustained timbres.

### 4.4.2 Sustained Timbre Synthesis for Composition

The results from the initial experiments led to the creation of a compositional utility that dynamically synthesized sustainable timbral textures. In this utility, the algorithms learned several different timbral textures that did not decay, essentially synthesizing new material and continuous transitions from a limited amount of data.

To test the utility's capability of synthesizing sustained timbres, I composed four different sonic timbres for training data. Each of the timbres was designed to not have a distinct attack or decay portion, yet be dynamic throughout its duration.

I trained a CRBM to synthesize each respective sonic timbre, resulting in 4 separate models. For the parameters of each timbral CRBM in this test, I set the STFT window size to 4096, the hop to 2048 (i.e. 50 percent overlap), and applied no tapering window. This resulted in a total of 2049 spectral features for each observation. From this, the internal parameter settings were set to be constant (i.e. 2049), the order of the model was set to 5 frames, and used 1 step of CD/Gibbs sampling. From these trained models, I synthesized audio by providing initialization data. For these models, I synthesized double the amount of sample frames I used for training (50 frames used to train each CRBM for 100 frames of synthesis), to test if the model could continuously synthesize the timbres, beyond the training set.

For classification, I created an ECOC(SVM) model using 6 (k=4) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 2000 STFT samples, evenly distributed across the classes.

The resulting output effectively extended the audio of the original samples and was classified with 95.75% accuracy (see 4.7). The synthesized audio did not deviate from the original timbral envelope, allowing for the continuous synthesis of material that sounded similar to the original samples and was dynamically generated for a specified duration.

Table 4.7: Resulting confusion matrix using ECOC (SVM) classification for synthesis of extended, sustained composed

| **Sustained Timbre** | Timbre A | Timbre B | Timbre C | Timbre D |
|---|---|---|---|---|
| Timbre A | 99 | 0 | 1 | 3 |
| Timbre B | 1 | 99 | 10 | 1 |
| Timbre C | 0 | 1 | 89 | 0 |
| Timbre D | 0 | 0 | 0 | 96 |

For a more dynamic example that took advantage of the previously tested aspects of the algorithm, I modeled a different composed timbre, Timbre C, which involved rapid changes in pitch across the sample. I synthesized approximately two, four, and twenty times the original length of the training sample. The resulting synthesis organically evolved the sound, ending with different timbres for each of lengths. The development of timbre that systematically shifted from the original sample into a new sound due to the generalizations learned by the algorithm bears resemblance to the networks of Bischoff, Tudor, (see 3.2.3) and the genetic algorithms of Xenakis (see 3.2.1 and [34]) and minimalist composers [55] [56]. This presents the composer with a utility that organically evolves sounds according to its own definitions, presenting altered material for investigation and inspiration.

The capability of CRBMs in this utility to dynamically synthesize new and evolving material, gives the composer a valuable asset to employ in their compositional design. I used this utility in several of my own artistic works (see 5) to generate continuous dynamic material.

### 4.4.3 Transitional Synthesis through Hidden Layer and Stylistic Label Manipulation

In all of the synthesis experiments, the theoretical construction of CRBMs and FCRBMs resulted in the algorithms learning hidden feature representations of the input data (see 2.2.3 and 2.2.4 for definitions). These representations are applied in the first two compositional utilities where new material is synthesized by the model through this abstraction, defining non-linear paths from one data class to another independent of the composer (beyond initialization and labeling).

When transitioning between timbres, FCRBMs use externally defined labels for direction, providing the composer with a high level method to control timbral synthesis without specifically defining it. This high-level composing through machines is a powerful capability, paralleling many of the advantages Xenakis sought with his computer works (see 3.2.1). I explored the possibilities of a composer 'piloted' system by modeling two different, distinct timbres using a FCRBM, creating a transitional algorithm that would synthesize both textures and a path between them.

In an initial test, transitioning between Timbre A and Timbre B, the algorithm generated transitions that sounded very similar to a cross fade. The two timbres were spectrally distinct from each other, with very little overlap timbrally. This lack of overlap led to synthesis where the sounds faded in and out as directed by the labels, without interference or the generation of any new sounds for the sonic transition.

This is also heard in a second test, transitioning between Timbre C and Timbre D. An interesting difference can be heard in the synthesized Timbre D after the transition. A spectral artifact is held over from Timbre C that is not present in the originally synthesized Timbre D, indicating that during the transition period, audio remnants and algorithmic parameters of Timbre C were present enough to become recursively aural in the new sound.

I explored this further in a third test where I modeled three separate sounds that had elements of overlap in their spectra: a 440 Hz sine wave (a single pitch, with no harmonics/overtones, based on the fundamental pitch A4), a 440 Hz sawtooth wave (a single pitch with all the harmonics/overtones having an amplitude of 1/harmonic number, based on the fundamental pitch of A4) , and Timbre C from the second utility (see 4.4.2).

Transitioning between the sine and sawtooth wave resulted in the respective addition (sine to saw) and subtraction (saw to sine) of the 440 Hz harmonics, although the transition from sine to saw resulted in slightly different amplitudes in the spectrum, altering the resulting timbre of the synthesized saw. This change of timbre through transitional synthesis is even more pronounced in the transition from Timbre C to the sawtooth wave. The more spectrally diverse Timbre C generated a larger imbalance in the sawtooth wave's harmonics, creating a completely different sound as a result of the autoregressive nature of the model. When sufficient energy is present in the spectral features, they are reinforced in the progressing timbres, regardless of whether they exist in the desired/labeled sound.

I further tested this capability through the construction of my final compositional utility: a singular FCRBM trained with 10 different timbres to be used for synthesizing each timbre and the transitions between.

I first evaluated the utilities capability to synthesize each of the timbres, using a similar testing process to 4.4.2. For the parameters of FCRBM in this test, I set the STFT window size to 4096, the hop to 2048 (i.e. 50 percent overlap), and applied no tapering window. This resulted in a total of 2049 spectral features for each observation. From this, the internal parameter settings were set to be constant (i.e. 2049), the order of the model was set to 5 frames, and used 1 step of CD/Gibbs sampling. From this trained model, I synthesized audio by providing initialization data and the desired label for each given timbre. For this model, I synthesized double the amount of sample frames I used for training (50 frames used to train each timbre of the FCRBM for 100 frames of synthesis of each timbre).

For classification, I created an ECOC(SVM) model using 45 (k=10) binary classifiers and a multi-class NB assuming an unbounded kernel distribution of the data, using a total of 5000 STFT samples, evenly distributed across the classes.

The resulting synthesis (synthesizing a singular class for a given number of frames) using this model achieved a 98.60% accuracy in an ECOC (SVM) classification test (see 4.8).

Table 4.8: Resulting confusion matrix using ECOC (SVM) classification for synthesis of straightforward synthesis in 10 class model

| **10 Class FCRBM** | Timbre A10 | Timbre B10 | Timbre C10 | Timbre D10 | Timbre E10 | Timbre F10 | Timbre G10 | Timbre H10 | Timbre I10 | Timbre J10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Timbre A10 | 99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Timbre B10 | 0 | 99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Timbre C10 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Timbre D10 | 1 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 1 | 0 |
| Timbre E10 | 0 | 0 | 0 | 0 | 98 | 0 | 0 | 0 | 0 | 0 |
| Timbre F10 | 0 | 0 | 0 | 0 | 1 | 99 | 0 | 0 | 0 | 0 |
| Timbre G10 | 0 | 1 | 0 | 0 | 1 | 1 | 100 | 7 | 0 | 0 |
| Timbre H10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 93 | 0 | 0 |
| Timbre I10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 99 | 0 |
| Timbre J10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

From this 10 class base, I explored varying transitions between the different classes of the algorithm, finding several interesting compositional possibilities beyond crossfades and harmonic imbalances of the initial test with three timbres. In one transition test, from a sine wave (Timbre A10) for 50 frames to a saw wave (Timbre B10) for 50 frames, the initialization of an exclusive frequency altered the weighted spread found in the saw wave, resulting in a brief period of oscillation, eventually settling on an imbalanced overtone structure favoring specific harmonics, and resulting in a new timbre.

By altering the number of frames to synthesize from each class, different initializations of the transitions and autoregressive paths were synthesized, creating new timbres dependent on where the transition occurred. I found in one transition test, where I explored transitioning from a chirp (Timbre E10) to a saw wave (Timbre B10) at different points across 100 frames of synthesized data, only subtle variations resulted in the harmonic amplitudes of the synthesized saw wave. Yet, in another test, transitioning from a chirp (Timbre E10) to a motoring, noisy timbre (Timbre F10) at different points across 100 frames of synthesized data, the resulting spectral changes were pronounced and quite drastic.

In the transition between a buzzy noise timbre (Timbre H10) for 50 frames and a chirp (Timbre E10) for 50 frames, aspects of the initial timbre were inherited by the second timbre, resulting in an effect similar to a filter, functionally merging the dominant frequency characteristics of Timbre E10 with the inharmonic spectrum of Timbre H10.

By alternating between two classes, I could create different timbral textures. By alternating between a saw wave (Timbre B10) and a choral timbre (Timbre I10), patterning the classes resulted in new sounds.

The variation across the synthesized timbres demonstrated the breadth and versatility of the FCRBM to synthesize and control complex spectral spaces. From the base synthesis resulting from manipulating stylistic labels and the hidden layer as an abstraction, a new compositional structure is presented from which the composer can create. By manipulating the labeling systems, thus the hidden layer, composers are able to access a new non-linear mapping system, which can be used to create transitions between timbres or new synthesized textures as a result of the model's learned abstraction.

## 4.5 Summary

In this chapter, I demonstrated how CRBMs and FCRBMs can be used to synthesize and control complex spectrum in a variety of contexts, gaining insight into how these algorithms work and how they best can be applied in music composition. I initially modeled timbres of musical instruments from the same musical family playing the same pitch, showing how the algorithms are able to synthesize subtle timbral differences, learning the ideal model parameters to perform this synthesis and the characteristics need for continuous timbral synthesis (i.e. sustained amplitudes). I then modeled different pitches, showing how more distinct timbral differences can be modeled with a higher accuracy, reinforcing the ideal model parameters found in the initial experiment and providing deeper insight into what combinations of timbres were ideal for synthesis. Finally, I created three sets of compositional utilities, exploring the creative capabilities of the algorithms to synthesize the segmented dynamic envelope of a sound, a variety of dynamic, sustained timbral textures, and new, machine-driven timbre via limited composer interaction. The implementation of these models in these varying contexts displayed the potential of CRBMs and FCRBMs for creating a systematic approach to timbral synthesis and control.

# CHAPTER 5

# APPLICATION IN ARTISTIC CONTEXTS

In this chapter, I describe two different compositional systems, implementing CRBMs and FCRBMs in artistic contexts, building upon the findings in 4. In section 5.1, I describe my work with *a performer's perspective*, defining a sonic choreography through the translation of dance movement. In section 5.2, I describe my installation series *is That('s) all there is*, where participants interact with pseudo-immersive, multi-modal feedback ecologies.

## 5.1 Generating Timbral Atmospheres Through Choreography in *a performer's perspective*

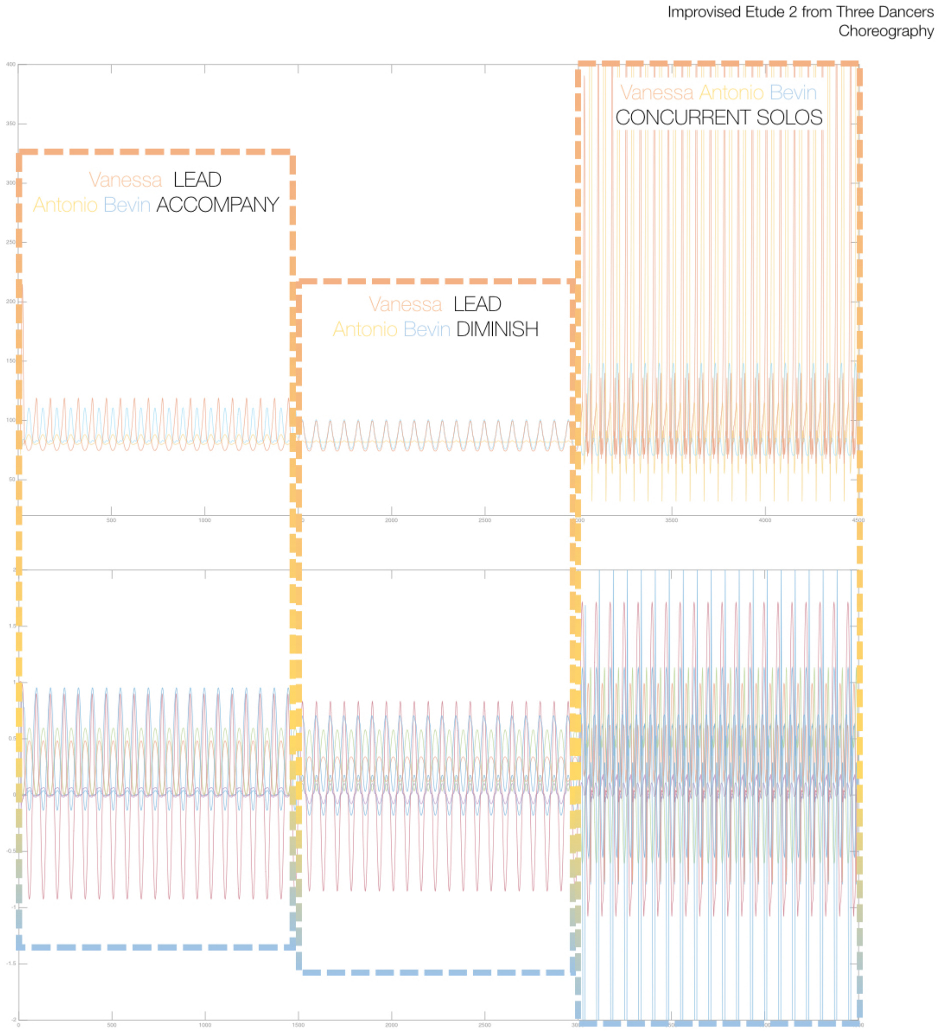


Figure 5.1: Score excerpt from *a performer's perspective*

*a performer's perspective* is an interdisciplinary dance project created by Shannon Cuykendall. In summarizing the project, Cuykendall states:

> We explore ways to transmit a dance performer's point of view through the creation of an interactive documentary. Using qualitative and quantitative research methods, we gathered a broad spectrum of data to understand the kinesthetic experience of dancers in Judith Garay's work, *the fine line twisted angels*. The dancers' data is presented through various forms and modes of

> interaction, providing audiences the opportunity to reflect, empathize and understand the choreography through multiple lenses. [57]

In this work, Cuykendall looks at how technology can “extend perceptions of the physical body and translate dance movement into new forms that transcend language [57],” seeking the interpretation of dance material by different artists and data scientists. I was given movement data of three dancers’ improvisations based on choreography established by Cuykendall and given an opportunity to make a composition based on the material from my perspective.

The data came in several streams: myo sensor measurements of acceleration, electromyography (EMG), and angular velocities; a Microsoft Kinect extrapolated skeleton frame; digital video from the Kinect and a GoPro. Given these varying sources of highly dimensional data in specific contexts (i.e. which dancer’s were performing when, in solo, duet, or trio, etc.), I decided to use FCRBMs to synthesize and control timbral material based on statistical measures of the dancer’s movement, wave form synthesis, and parameter definition of an array of granular synthesizers, driven by a metaphoric choreography defined by the algorithms and guided by the formal decisions of the composer. In order to create the compositional system to synthesize and control timbre in this way, I separated it into three parts: creating generalized *dancer representations* of each dancer, generating source control data and audio data using waveform synthesis based on measured statistics from the sensor data; training a FCRBM for *timbral texture choreography* using the audio from the dancer representations to synthesize timbres representing each dancer and transitions between these timbres; training a second FCRBM for *granular parameter choreography* using the control data from the dancer representations to generate control parameters for the granular synthesizers.

I created separate statistical representations of each of the three dancers from the initial data streams to create the *dancer representation*: a control data stream based on normalized measures of the dancer’s improvisation and source audio material generated from a combination of statistics at different time rates through waveform synthesis (see 5.2). For each respective dancer, I created a statistical representation of their movement as gathered by the sensors, taking a combination of windowed means and variances to measure the respective dancer's improvisation. This statistical representation was used as source *control data* to be learned by an FCRBM to drive granular synthesis. These statistical measures were also used to create waveforms through a variety of frequency and amplitude modulations, translating dancer movement into a statistically representative sonic timbre. This waveform was used as source *timbral data* to be learned by a FCRBM to synthesize timbre. By creating isolated control and timbral representations of each dancers' actions, I was able to source parameter controls and sounds directly reflecting the movement of the dancers, translating their improvised movement into audio.

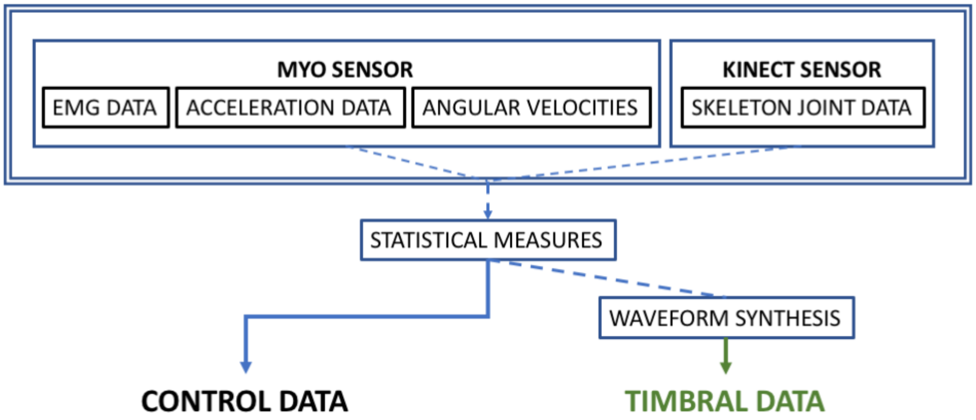


Figure 5.2: Diagram of the transformation of sensor data into *dancer representations* of control and timbral data

The resulting timbral data from each *dancer representation* was used to train a FCRBM from which continuous, dynamic timbral combinations and transitions could be algorithmically synthesized (see 5.3), resulting in the *timbral texture choreography* synthesis module of the system. Each dancers' *timbral data* was labeled and learned by the FCRBM, giving the composer the ability orchestrate the dance ensemble, balancing and mixing the timbral synthesis as a choreographer would direct group movement. Thus, the source materials for the larger compositional form is chosen by the metaphoric compositional choreography, creating sonic dancers and compositional directions from the initial improvisation material.

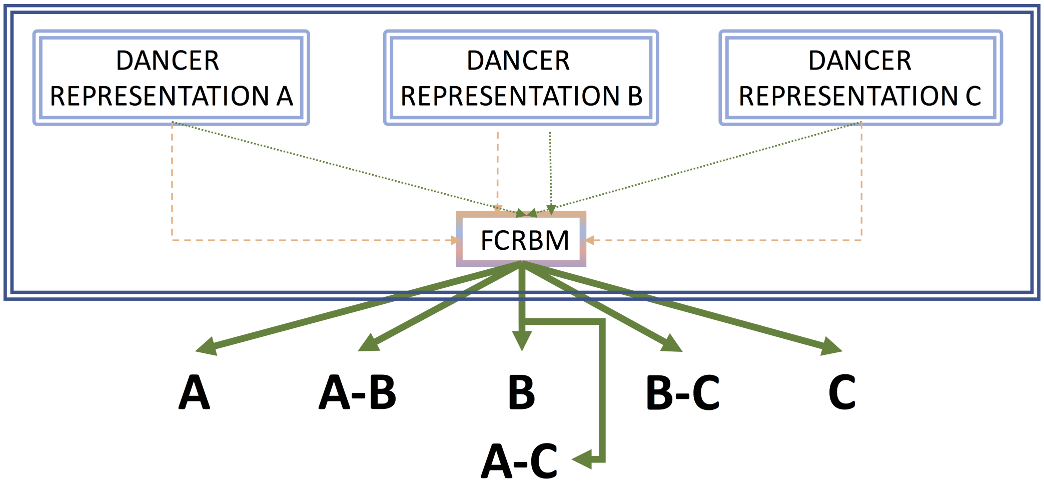


Figure 5.3: Diagram of using timbral data (green dotted lines) and composer defined labels (orange dashed lines) to synthesize machine-driven timbres resulting in the *timbral texture choreography* module.

In addition to this *timbral data*, I also took the *control data* from the *dancer representation* and defined choreographic generalizations that the composer could apply to an array of granular synthesizers [58] (see 5.4), resulting in the *granular parameter choreography* control module of the system. This normalized control data was learned by a FCRBM, which synthesized control data and transitions based on composer defined labels. These synthesized control parameters served as another vehicle for metaphoric choreography, creating representations of the different dancers, that could be used as a mapping schema for the next part of the module. This metaphoric choreography was then mapped to a set of granular synthesis parameters using another FCRBM, fully connecting the resulting granular synthesis to the composer's chosen choreography.

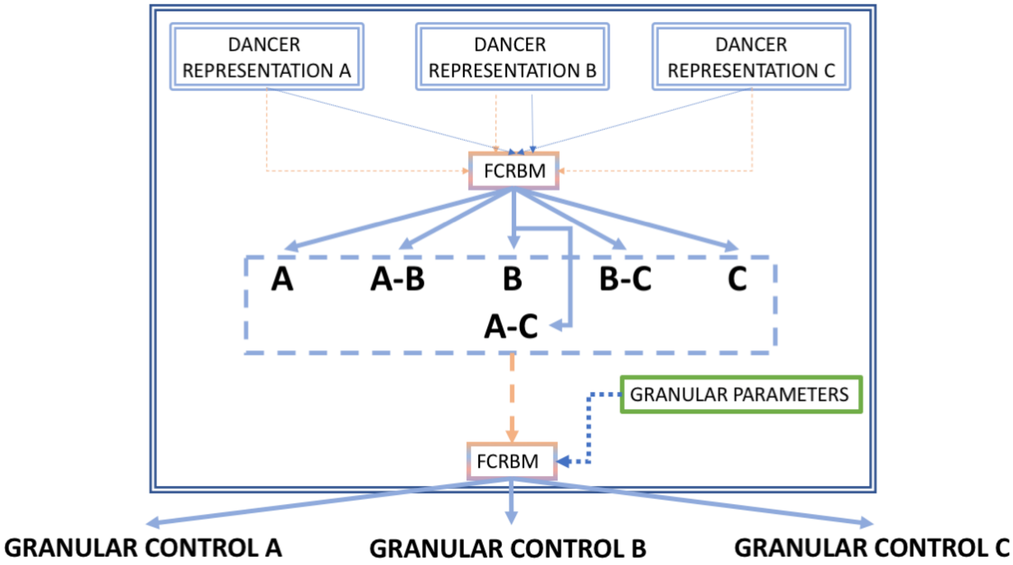


Figure 5.4: Diagram of using control data (blue dotted lines) and composer defined labels (orange dashed lines) to synthesize control data, then mapping that control data (larger orange dashed line) to granular synthesizer parameters (larger blue dotted line) resulting in the *granular parameter choreography* module.

The resulting compositional system allows artists to compose sonic atmospheres through choreographic metaphor (see 5.5), creating dynamic sonic atmospheres that reflect dancers' performance. The system takes choreography and translates it into timbral manipulations of constructed waveforms based on the dancers' initial improvisations, generating sounds that it learned in training, providing the composer with a new, machine-driven method of generating aural consequence through the lens of choreography.

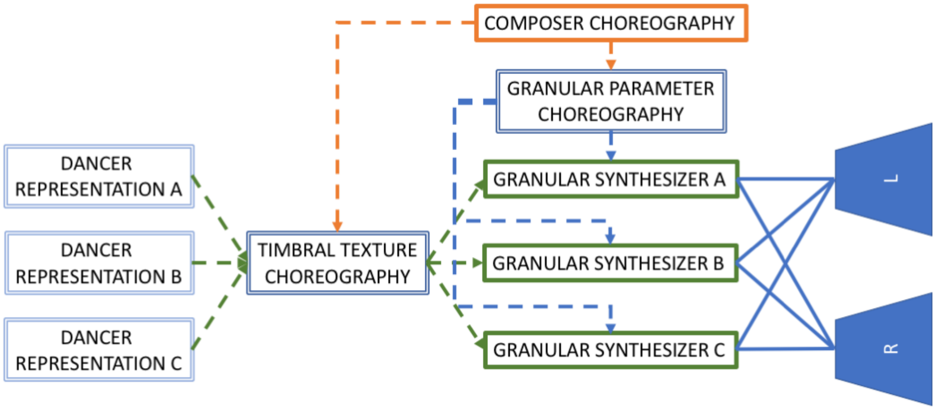


Figure 5.5: Diagram of the complete compositional system architecture for *a performer's perspective*

This system is a realization of compositional influences through FCRBMs: Xenakis' composer as a pilot (see 3.2.1), directing control through dynamically synthesized representations based on dancer's movements; Saariaho's exploration of timbral expansion (see 3.2.2) through the extension of already present contexts, providing the composer with sonic textures that would adapt based on their decisions; Bischoff and Tudor's multi-layered networks (see 3.2.3), creating an interconnected control and timbral synthesis system, reliant on each of its components to realize the composition.

To demonstrate this synthesis in application, I wrote three etudes [59] using the system. The resulting timbral synthesis is a dynamic sonic space that is easily translatable to more complex forms using choreographic metaphor. The development of this work inspired a deeper investigation into comprehensive model systems, which factor not only the internal elements of the algorithms, but connect separate models to each other (see 6.3.1).

## 5.2 Dynamic, Multi-Modal Audience Engagement in *is That('s) all there is*

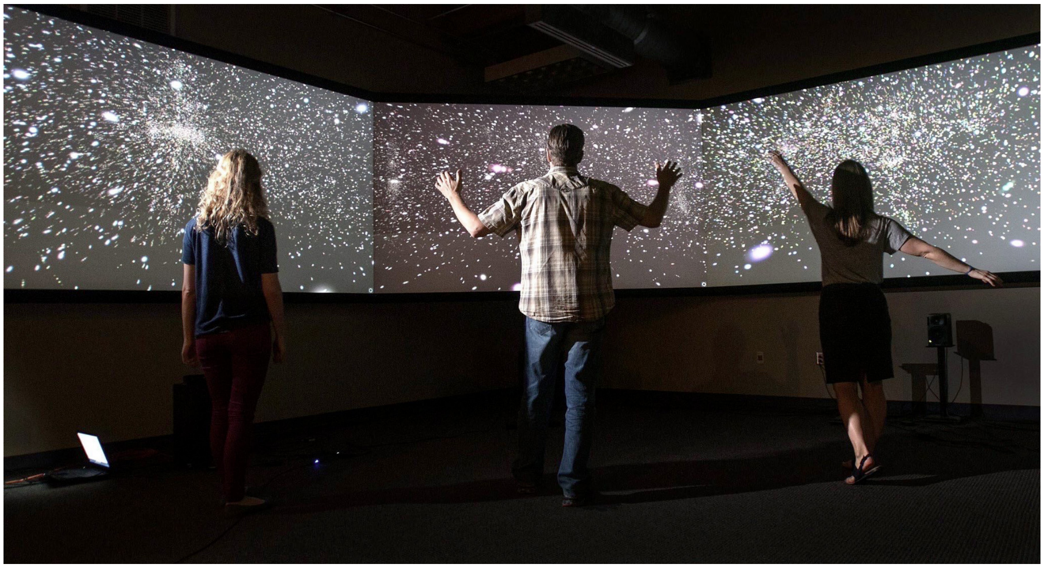


Figure 5.6: User's interacting with *is That('s) all there is*

In the installation series, *is That('s) all there is*, I created three multi-media art installations at three separate events exploring the same venue: *,you sound so familiar...*(2016), *to me..., as if i've heard this before*(2016), and *...somewhere, right now*(2017).

These installations incorporated observer's movement into an interactive visual and sonic atmosphere, creating a dynamic, feedback artistic ecology. By gathering and analyzing observers' movement features, elements of the installation 'reacted' to the observers, relaying imagery and audio throughout the environment. As the observer moved and interacted within the space, they became integrated directly with the visual and aural elements, creating a responsive feedback ecology, embracing the approach and concept of system aesthetics (see 3.1).

The ecology included three main elements: movement analysis, audio synthesis, and visual synthesis.

- For movement analysis, the installation three of Microsoft Kinect V2 to gather and pipe skeleton features to custom designed software where movement analysis was performed in real-time. This analysis was translated into control data, which directed the audio and visual synthesizers.

- For audio, the installation used automated, granular synthesizers, generating probabilistic responses to the movement analysis, setting the parameters of the synths within stochastic response ranges, realized in 10.1 surround sound.

- For the visuals, the installation used a cluster of video synthesizers projected onto three screens, generated by analyzing the spectrum of the audio generated by granular synths.

In order to control and drive these elements, I relied on two separate modules that used CRBMs and FCRBMs: *timbral texture module* and *control parameter module*.

The *timbral texture module* (shown in 5.7) was used to generate source timbral textures offline used by the granular synthesizers in the ecology. Using different audio samples created by analog electronic circuitry, three CRBMs were trained to synthesize extended timbral textures. These textures were labeled and used to train a FCRBM, which synthesizes timbres and spectral transitions. The ultimate output of the module is three dynamic timbres generated by the respective CRBMs (A, B, and C in 5.7) and six transitional textures (A-B, A-C, and B-C in 5.7).

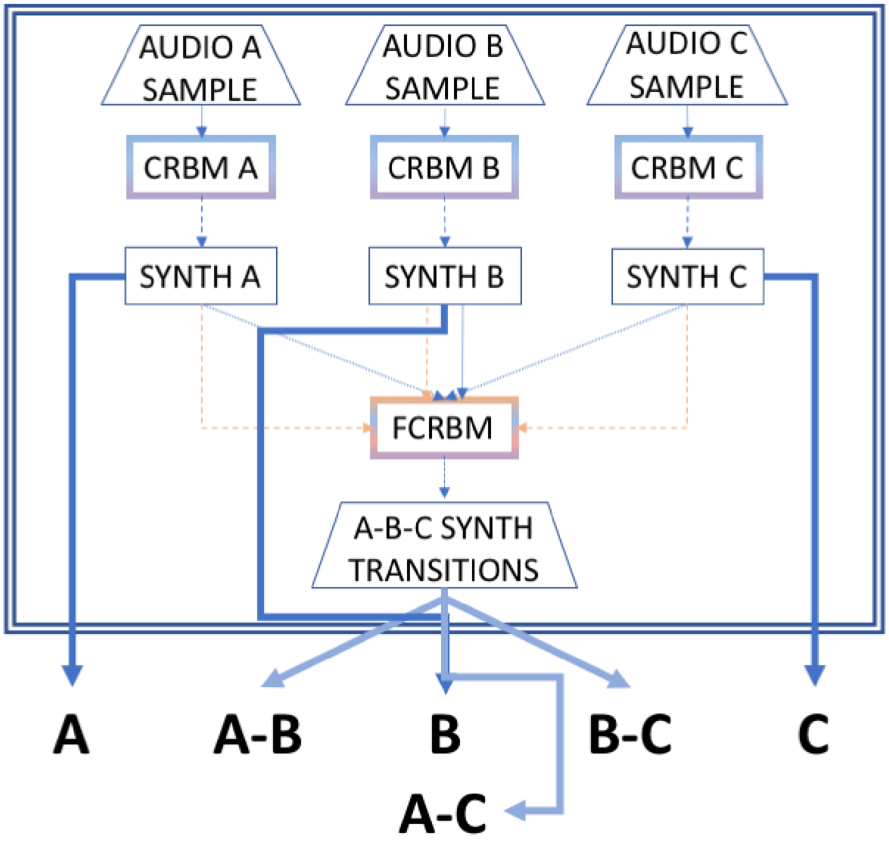


Figure 5.7: Diagram of *timbral texture module*. Audio samples are learned by CRBMs, which produce synthesize timbres (dark blue solid lines) and train a FCRBM to learn transitions between those timbres (light blue solid lines) as directed by composer labeling (orange dashed lines).

The *control parameter module* (shown in 5.8) is used to drive the control parameter sets of two stochastic-range granular synthesizers. The stochastic-range granular synthesizer is a synthesizer that take ranges of 5 different parameters (i.e. grain rate, grain length, grain pitch, grain amplitude, source audio location) and generates a randomly selected feature set from those ranges continuously for granular synthesis. For these ranges, I defined three separate control parameter sets, labeled them, and trained a FCRBM to synthesize these sets and transitions between them. The ultimate output of the module is three static granular control parameter sets and six transitions between each of those static parameter sets, used to drive the granular synthesizers relative to the corresponding labels.

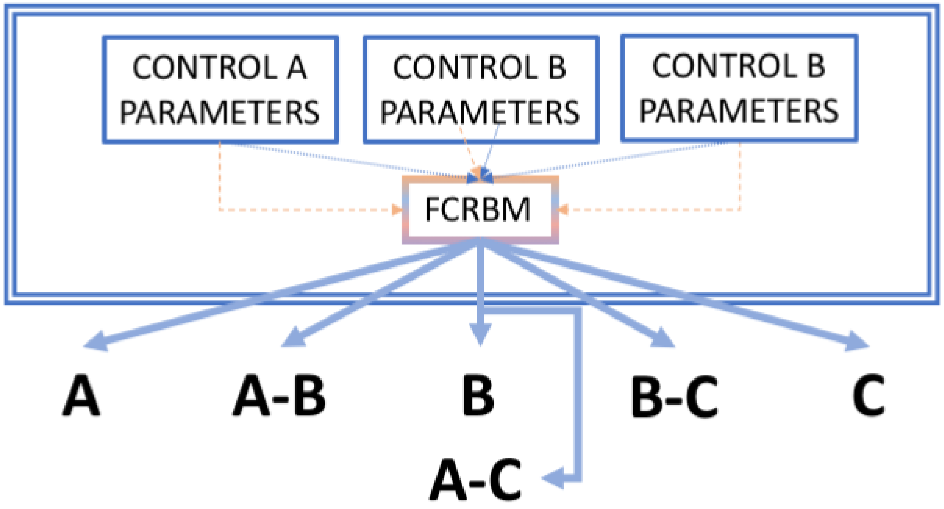


Figure 5.8: Diagram of *control parameter module*. Composer determined parameter sets (blue dotted line) and composer determined labels (orange dashed lines) are used to train a FCRBM to synthesize granular control parameter sets and the transitions between those sets.

The resulting modules were then used in the larger sonic ecology architecture (see 5.9). The granular synthesizers were sourced with the timbral textures that were created using the *timbral texture module* before real-time, observer interaction with the ecology. In order to access the different timbres, the source audio location parameter of the granular synthesizer was used to isolate specific sonic targets during the interactions (i.e. all audio was collected in one sound file and specific points of that audio were used for granular synthesis). The *control parameter modules* were driven by an aggregate movement analysis and recognition algorithms that recognized different classes of movement within the ecology based on data gathered from the Microsoft Kinects (V2).

The movement analysis and recognition algorithm analyzed group movement features (i.e. the number of active bodies present in the ecology, their relative positions, their average 'energy,' and the amount of time spent interacting with the ecology) and made a probabilistic classification across three predefined classes using a hierarchical hidden-Markov model [60] in real-time. This probabilistic label was sent to the *control parameter modules* to drive the FCRBM synthesis of control parameters for granular synthesizers. After the resulting synthesis, the sound was diffused across an 10.1 surround sound system, according to an externally composed algorithm for each timbre.

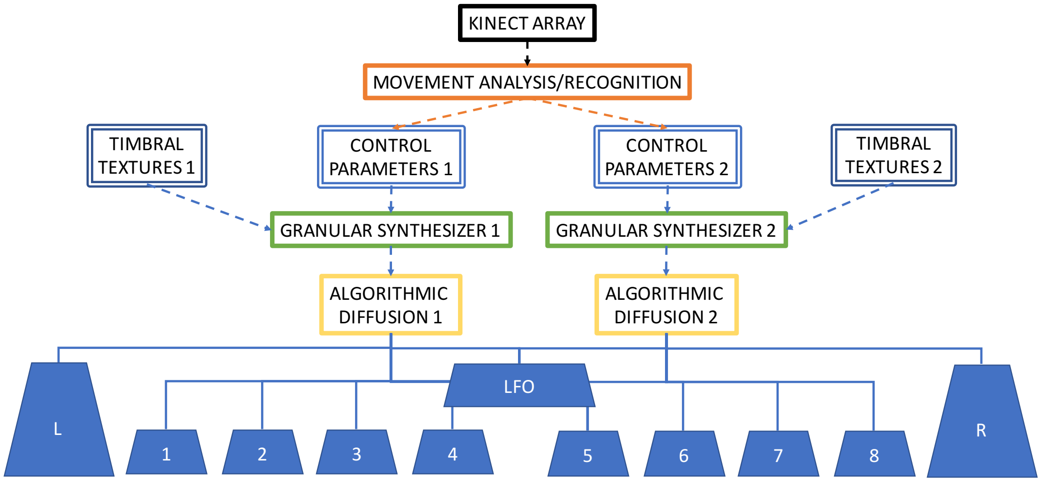


Figure 5.9: Diagram of *is That('s) all there is* sonic ecology

The resulting multi-modal ecology is a culmination of the utilities and research developed from this dissertation. The synthesized timbres and control values are a direct product of the compositional utilities developed in 4.4 (specifically see 4.4.2 and 4.4.3). The full integration of observer, composer, and algorithm into one cohesive aesthetic system is a realization of the concepts initially explored in system aesthetics (see 3.1, specifically 3.1.1) and expanded by composers (see 3.2). The integration of systematic and relevant musical ideologies with algorithmic design and architectural development realizes the capability of CRBMs and FCRBMs to synthesize and control complex timbral representations via music composition systems.

## 5.3 Summary

In this chapter, I described two compositional systems, *a performer's perspective* and *tis That('s) all there is*, realizing the full capabilities of CRBMs and FCRBMs in facilitating complex timbral synthesis and control in artistic contexts. The implementation of these systems is a clear progression from the integration of the work of past machine-learning scientists (see 2.1), artists (see 3), and the results of experimentation with the algorithms (see 4).

# CHAPTER 6

# CONCLUSION AND FUTURE DIRECTIONS

The work being done with algorithmic design and implementation in the field of music composition is rich with possibility. In 6.1 I summarize the progression of my research, describing the context, motivation, and implementation of using CRBMs and FCRBMs to synthesize and control timbre. In 6.2, I describe additional technical capabilities these algorithms possess, providing composers with a strong motivation for future use when working with digital timbral synthesis. In 6.3, I explain how the theory of CRBMs and FCRBMs lend themselves especially well to continuing to address formalistic and aesthetic problems in music composition. In 6.4, I discuss how composers can build upon these algorithms for creative expression through the development deep belief nets, extending their practice into non-linear interactions, multi-modal mapping applications, and higher levels of abstraction. Finally, in 6.5, I conclude with the larger effect of CRBMs and FCRBMs on musical synthesis, showing how through the implementation of these algorithms provide composers with a human-machine method to advancing their aesthetic.

## 6.1 Demonstrated Research Summary

The use of machine-learning technology in musical synthesis applications offers composers new and novel methods of developing their own aesthetic, as established in the application of the algorithms in audio and movement domains (see 2.1). Artists (see 3.1, 3.1.1, and 3.1.2) and composers (see 3.2) have already created a foundation for using technological systems creatively in order to advance their own intent. CRBMs and FCRBMs provide a theoretical and practical base for exploring the use of algorithms for advancing musical composition, specifically in the area of timbral synthesis and control. This is evident in the implementation of the algorithms to accurately synthesize unique instrumental timbres (see 4.2), pitches (see 4.3), the dynamic envelope of sonic events (see 4.4.1), and continuous, dynamic textures (see 4.4.2). Through these implementations, compositional systems have been developed that facilitate aural realizations of machine agency, empowering composers and expanding their aesthetic (see 5). My research has demonstrated that these algorithms possess the capabilities to create new opportunities for timbral synthesis and control.

## 6.2 Technical Advantages Presented by these Models

CRBMs and FCRBMs by their very construction present several computational and technical advantages to the composer, specifically dimensionality reduction, developing efficient synthesis via algorithms, and the generation of audio material that is representative of the sound it models, yet continuously dynamic and unique.

### 6.2.1 Dimensionality Reduction and Computational Efficient Synthesis

By creating an abstract representation of the data, CRBM and FCRBM can reduce the dimensionality of a dataset and still synthesize representative data accurately (see 4). Through the generalization of data to a hidden unit layer, the dimensionality of the data is reduced to the number of hidden units in the layer. When synthesizing new material, these algorithms only need the weights and biases that connect the internal parameters of the model to synthesize new material.

From these components, the algorithms can synthesize dynamic timbres of traditional instruments and composed sounds (see 4). In the demonstrated experiments, the CRBM was able to use a reduced hidden layer abstraction to accurately synthesize a timbre with a much higher dimensionality. Specifically, the reduced hidden units setting (512 hidden units) was able to generate a spectral reconstruction (4096 features) with a very high accuracy (higher than 95%). The contexts in which this was successful included modeling the unique instruments of the strings (95%), woodwinds (96%), and pitched percussion (96%), and modeling the different pitches of a violin(98%), an oboe

(94%) and bells (100%). When working with compositional timbres, CRBMs were able to achieve accuracies of higher than 96% using a constant set of features (2049 hidden units).

While not performing as high as the CRBMs in the explicit synthesis experiments (i.e. modeling a specific instrumental sound for a specific duration), the capability of the FCRBM to encapsulate the characteristics of several different timbres within the same model gives composers another technically efficient utility for synthesizing dynamic timbres (see 4.4.2 and 4.4.3). Instead of training several different models for the same task (i.e. 8 CRBMs to model the 8 scale steps), a composer can train a single model (i.e. one FCRBM with 8 classes to model the 8 scale steps), in fewer iterations, making the task more efficient in implementation.

In these cases, when the timbre being synthesized was sustained, data could be generated on demand by the algorithm, presenting a much more efficient method of manufacturing large amounts of data for other tasks.

### 6.2.2 Synthesizing Dynamic and Transitions Timbres

Beyond dimensionality reduction and computational efficiency, CRBMs and FCRBMs provide composers with a method to synthesize continuous, dynamic textures at will. Timbres that would be costly and difficult to reproduce, either due to the need for specialists/instrumentalists or the very nature of the material being generated (e.g. stochastic analog circuitry, improvisation) could be generated by a composer to their desire, as seen in the timbral synthesis of the compositional utilities (see 4.4) and the music composition systems (see 5) I developed. These algorithms provide a way to synthesize dynamic timbres that are similar yet not exact reproduction of the original data, providing a richer pallet from which to compose.

The ability to synthesize timbres and factorize their connections using FCRBMs enable the composer to synthesize transitions between several different timbres within a singular model (see 4.4.3)). This multi-class transitional synthesis model derives its own method of synthesis from the guidance of the composer, acting as a collaborative agent in the compositional process, fully realizing concepts of system aesthetics (see 3.1). Several of the models constructed for this purpose synthesized varied and interesting timbres that were not present in the original training samples, opening these systems to deeper aesthetic explorations (see 4.4.2 and 4.4.3).

## 6.3 Aesthetic Expansion via Machine-Learning Theory

The compositional and aesthetic opportunities provoked by the successful implementation of CRBMs and FCRBMs in music composition systems is myriad and rich for exploration.

### 6.3.1 Composer Empowerment Through Algorithmic Factorization

The ability to factorize interactions between the various components of FCRBM models empowers composers with the ability to define and direct synthesis from alternative modes. These algorithms can be coupled with external multi-modal utilities such as gesture/movement recognition algorithms, making mapping schemas that can be defined from a higher level (see 5.2). In *is That('s) all there is*, this resulted in fully immersive and intuitive interaction spaces where observers were able to move freely and focus on the consequences of their actions, rather than their performance of a limited vocabulary of interactions.

This factorization also addresses the data overload that can stunt creative expression when attempting to synthesize and control complex material such as musical timbre. The increasing complexity of digital interactions and software utilities often require composers and artists to adopt entirely new domains (i.e. computer science) or commit immense resources to realize effective performance. In music composition, this is most noticeably demonstrated in the instrumental performance of compositions from the New Complexity [61] [62]. The ability to expand compositional expression and sonic exploration through a mapping schema that does not require a performer or composer to sacrifice the expertise of their own domain would be hugely beneficial to expanding aesthetics and thought in musical composition, providing more opportunities to interact with complex musical paradigms, such as timbral synthesis and control. FCRBMs are capable of providing such a mapping schema, given further research.

### 6.3.2 Mapping Performance to Non-Linear Structures

One of the most intriguing opportunities motivated by the experimental results of my research was the ability to map internal model parameters directly to instrumental control systems, potentially making a synthesizer that could be cued through the manipulation of the algorithm's parameters. In CRBMs, composers can transform machine synthesis through the manipulation of the hidden layer activation distribution. After learning the model, composers can bypass the visible layer input and activate the distribution of hidden units directly, synthesizing data from a binary abstraction and creating non-linear perspective of control. In FCRBMs, where a variety of timbres could be generated from labeled input alone, the composer could directly trigger dynamic timbres by changing labels.

A proposed future interface that explores these different methods of algorithmic synthesis could be similar to a MIDI instrument, allowing composers and musicians to change the algorithms' parameters, whether they be labels or hidden unit activations, directly in real time. Each mapping schema would be direct yet provide two vastly different ways of exploring timbral synthesis, providing non-linear gateways into dissecting sound.

I have begun to explore this potential in *Improvisation for Vibraphonist and Network* (2017-2018), a telematic collaborative composition resulting from the interaction between human agent and machine system. Using attributes of latency, processing variability, and performer direction through the abstracted hidden layer and labels of an FCRBM, I've created an improvised network that incorporates performative, non-linear elements into synthesis. The composition uses the stylistic labeling of the FCRBM to map a MIDI pitch organization from an electronic vibraphone to the timbral synthesis of the algorithm. This provides the vibraphonist with the ability to explore textural accompaniments to their own improvisations, without having to abandon any of their performative practice.

A natural extension of this enhanced control paradigm is the development of efficient multi-modal interactions. The translation of modes across a common algorithmic vernacular (as Xenakis anticipated, see 3.2.1) provides a creative system design that defines connections and interactions between agents rather than their exclusive outcomes, as realized between user movement and algorithmic sound synthesis in *a performer's perspective* (see 5.1).

Composers can delve deeper into mapping schema and generalized stylistic connections of the models, directly exploring the non-linear structures of the algorithms. Exploration of this space provides access to innumerable combinations, each resulting in their own unique timbre. Navigating across these timbres provides completely new methods of timbral deconstruction, filtering, control, and synthesis.

## 6.4 Deep Learning Applications

Researchers have been able to efficiently train deep belief nets (DBN), using multiple layers of CRBMs to learn continuous variables [63] [64]. Delving even further into the formal components of the CRBM theory, these stacked layers provide insight into the construction of the machine-learned connections, giving access to the elements of these patterns, opening them to manipulation and further investigation. Such architectures generate deep feature sets that are capable of a variety of applications such as modeling human motion [65] [14], phone recognition [46], and acoustic modeling [47].

I have begun to explore deep learning application in musical timbre synthesis (see C.1). In these explorations, the dFCRBM is a 2-layered net, with each layer being a FCRBM. In addition to providing a more developed abstraction of the spectral representations, this additional layer also gives the composer another set of labels and hidden units that can be manipulated and organized for compositional purposes.

The deep learning architectures leave much to be desired as shown in their performance with the experiments I used to validate the CRBM and FCRBM (see C.1), but deep belief nets offer intriguing compositional capabilities, as evident in the unique synthesis of the dFCRBM, the deeper, more complex control structure of the network, and other successful implementations already done using CRBMs [64] [66] [63] [46] [47].

## 6.5 Final Thoughts

Machine-learning provides a path toward more powerful methods of mapping and data synthesis. Composers can use these methods for creative expansion, as demonstrated in this research's application of CRBMs and FCRBMs to synthesize and control timbre . Micro-control aspects of composition, such as defining the transitional interpolations between timbres and parameters or generating unique, dynamic timbral material, can be relegated to the algorithms. This liberates the composer's process and allows for the focus on macro-concepts aspects of their aesthetics, of intention and expression. This fusion of composition with technological enhancement creates a codependence that provides a direct path for creating music composition systems that can achieve higher levels of expression.

From a larger view, the ability to compose from a system perspective that incorporates human and machine agency, opens the composer to a new form of creation that directly integrates aesthetic vision with contemporary technological thought. The composer is forced to analyze their process through algorithmic interaction, the use of abstract compositional systems, and their role as a human-agent within an open system. This analysis leads to a more developed aesthetic, resulting in more informed compositional choices that consider the design and application of the compositional systems that they use.

# APPENDIX A

# TECHNICAL CONSTRUCTION DETAILS

Working with immense multi-dimensional datasets forced me to consider computational efficiency when implementing the algorithms. Through this investigation, I found several opportunities to optimize performance and learning with the algorithms, specifically moving learning from serial processing on CPU to parallel processing on the GPU and optimizing stochastic gradient descent.

## A.1 Graphic Processing Unit (GPU)

The bulk of the processing in the FCRBM occurs in several simple matrix operations on large feature sets. While a CPU is only able to utilize a few cores in serial for this process, a GPU is designed to process a multitude of smaller tasks simultaneously. Through the development of CUDA [67], a high level parallel computing platform and programming model, code can be sent straight to the GPU from the CPU. Abstractions of this method have led to the implementation of many previously CPU implemented programs and code to run much more efficiently on the GPU [68], including an implementation in Matlab [69]. This ultimately is a much more cost and computationally efficient method for this research compared to parallelizing the cope on multiple CPU cores.

In order to validate the efficiency of moving these algorithms from the CPU to the GPU, I constructed a simple test. I modeled a singular dynamic timbre using a FCRBM trained with spectral data derived from a 20 second audio sample. I performed a STFT on the audio data using a STFT window size of 4096, a hop of 2048 (50% overlap), and did not apply a tapering window, resulting in 429 spectral samples. For the internal model parameters of the FCRBM, I used 2000 hidden factors, hidden units, and hidden features, set the order of the model to 5, and used one step of contrastive divergence. Using this data with these model parameters, I trained the same FCRBM algorithm for 5000 epochs, processing on the CPU in the first instance and on the GPU in the second, timing it using Matlab's internal profiler. The CPU I used was an *Intel(R) Core(TM) i7-6700K CPU @ 4.00 GHz*, which has 4 cores on a machine that had 32 GB of memory. The GPU I used was a NVIDIA GeForce GTX 1080, which has 2560 CUDA cores, on the same machine.

In performing the same task using the same data, GPU usage dramatically improved learning run-time (A.1).

Table A.1: Resulting runtimes of CPU and GPU in validation task

| Processing Unit | Total Run Time(min) | Approx Time per Epoch (sec) |
|---|---|---|
| CPU | 763.46 | 9.16 |
| GPU | 44.52 | 0.53 |

In looking more closely at where the majority of the time was spent in processing, it was clear the FCRBM would perform better on the GPU. The top 5 most costly operations in the code, accounting for approximately 34.7 % of the algorithm's run-time, all involved large element-wise operations (i.e. element-wise multiplication or division in combination with other operations) on large multidimensional datasets (i.e. the autoregressive connections to 'past' data). For these specific tasks, it was evident that the GPU's parallel processing was advantageous to the CPU's serial process.

## A.2 Optimizing Stochastic Gradient Descent

In creating models to synthesize audio, a large and diverse feature set was used. This feature set often created very volatile conditions for learning via stochastic gradient descent (SGD), frequently resulting in models becoming unstable and not learning parameters that could be used for synthesis. In order to find a better method to learn models with greater stability, I investigated different ways to optimize and control SGD. While previous FCRBM models [14] have utilized momentum [70], several other approaches have been used to optimize SGD including a similarly static optimization like the Nesterov accelerated gradient [71], and adaptive learning rate optimizers such as Adagrad [49] and RMSprop [72].

Without any attempt to optimize SGD, the updates to a FCRBM's parameters ($\theta$) in learning would simply be updating the parameters at the previous timestep ($\theta_{t-1}$ in the opposite direction of the gradient a given function, given a learning rate ($\eta$) set between 0.0 and 1.0:

$$\theta_t = \theta_{t-1} - \eta \times \Delta\theta \tag{A.1}$$

When using momentum, the update adds a user determined portion of the previous update ($\gamma$) to the current timestep's update in an attempt to drive the learning toward convergence:

$$v_t = \gamma v_{t-1} + \eta \times \Delta\theta \tag{A.2}$$

$$\theta_t = \theta_{t-1} - v_t \tag{A.3}$$

The Nesterov accelerated gradient builds on the momentum method, calculating the gradient on the anticipated future position of the parameters, essentially correcting the updates at each timestep with respect to the approximated future parameters:

$$v_t = \gamma v_{t-1} + \eta \times \Delta(\theta - \gamma v_{t-1}) \tag{A.4}$$

$$\theta_t = \theta_{t-1} - v_t \tag{A.5}$$

Adagrad adaptively updates each parameter with a different learning rate depending on frequency of the parameter's contribution to the cost of the function, eliminating the need to manually set the learning rate. It does this by adjusting the learning rate by the sum of the squares ($G_t$) of all the previous gradients (plus a smoothing term of a respective parameter

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} \times \Delta\theta \tag{A.6}$$

RMSprop focuses Adagrad, taking only a specified window of past gradients to temper the learning rates:

$$E[\Delta\theta_t^2] = \gamma E[\Delta\theta_{t-1}^2] + (1-\gamma)\Delta\theta^2 \tag{A.7}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{E[\Delta\theta_t^2] + \epsilon}} \times \Delta\theta \tag{A.8}$$

In order to evaluate the efficiency of these optimizers in application, I set up a test comparing the performance of each method by looking at the convergence of the measured error (i.e. the difference between the actual data and its reconstruction) and the resulting output.

I modeled a singular dynamic timbre using a FCRBM parallelized on the GPU and trained with spectral data derived from a 20 second audio sample. I performed a STFT on the audio data using a STFT window size of 4096, a hop of 2048 (50% overlap), and did not apply a tapering window, resulting in 429 spectral samples. For the internal model parameters of the FCRBM, I used 2000 hidden factors, hidden units, and hidden features, set the order of the model to 5, and used one step of contrastive divergence. Using this data with these model parameters, I trained five separate FCRBM algorithms for 5000 epochs, testing performance using no SGD optimizer, momentum, the Nesterov accelerated gradient, Adagrad, and RMSprop.

In implementing each of these optimization algorithms, Adagrad provided the most improved and stable learning.

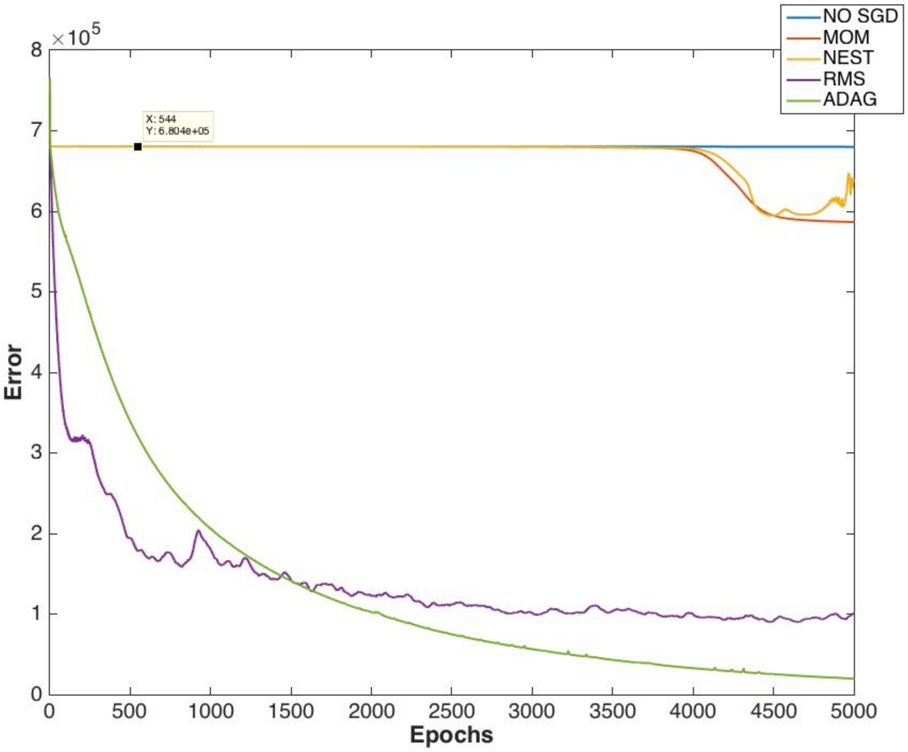


Figure A.1: Error of SGD Optimizers over 5000 epochs

While both Adagrad and RMSprop had typically learning curves as they reduced the error in the model, the performance of no optimizer, momentum, and the Nesterov accelerated gradient were forced into atypical curves in order to prevent instability in learning. In order to maintain stability, the learning rates of the non-adaptive optimizers had to be dramatically reduced, preventing the model from oscillating out of control. This shows an additional advantage to optimizers using adaptive learning rates, as they are able to defined and maintain stability throughout the course of learning.

# APPENDIX B

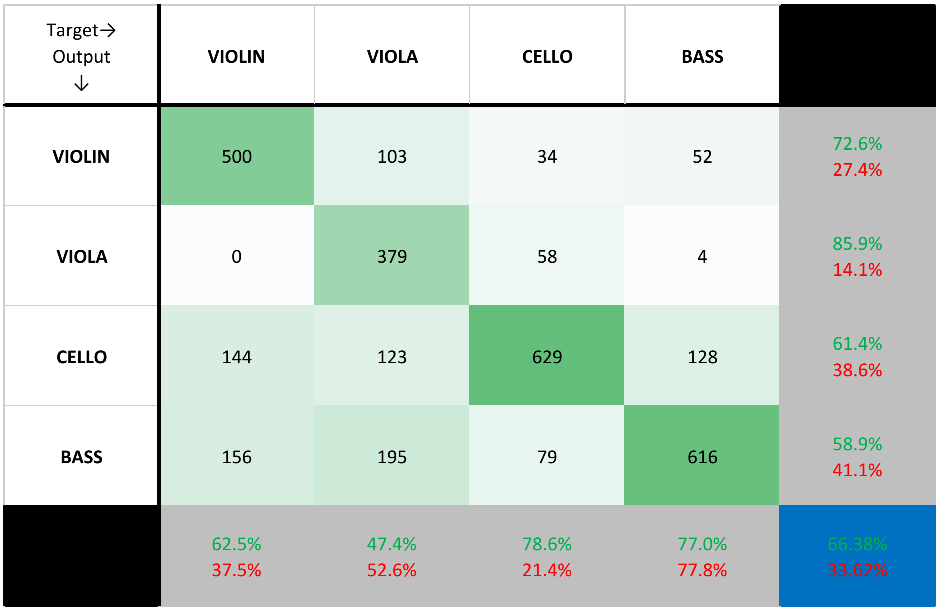
# RESULTS OF VALIDATION EXPERIMENTS

## B.1 Complete List of Tested Models

Table B.1: Names and parameters of tested models

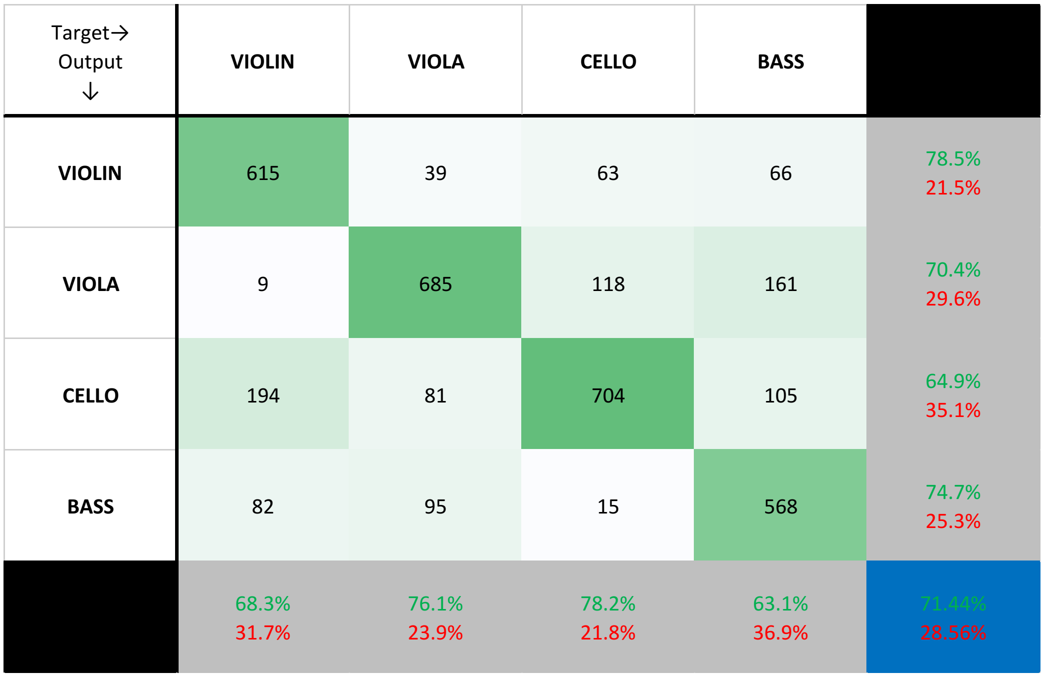
| MODEL | ALGORITHM | INTERNAL PARAMETERS | ITERATIONS |
|---|---|---|---|
| | | (numHid, numFac, numFeat) | (numCD, numGibbs |
| $C_{r1}$ | $CRBM_{sep}$ | redDim | 1 |
| $F_{r1}$ | $FCRBM$ | redDim | 1 |
| $C_{c1}$ | $CRBM_{sep}$ | constant | 1 |
| $F_{c1}$ | $FCRBM$ | constant | 1 |
| $C_{a1}$ | $CRBM_{sep}$ | augmented | 1 |
| $F_{a1}$ | $FCRBM$ | augmented | 1 |
| $C_{r5}$ | $CRBM_{sep}$ | redDim | 5 |
| $F_{r5}$ | $FCRBM$ | redDim | 5 |
| $C_{c5}$ | $CRBM_{sep}$ | constant | 5 |
| $F_{c5}$ | $FCRBM$ | constant | 5 |
| $C_{a5}$ | $CRBM_{sep}$ | augmented | 5 |
| $F_{a5}$ | $FCRBM$ | augmented | 5 |
| $C_{r20}$ | $CRBM_{sep}$ | redDim | 20 |
| $F_{r20}$ | $FCRBM$ | redDim | 20 |
| $C_{c20}$ | $CRBM_{sep}$ | constant | 20 |
| $F_{c20}$ | $FCRBM$ | constant | 20 |
| $C_{a20}$ | $CRBM_{sep}$ | augmented | 20 |
| $F_{a20}$ | $FCRBM$ | augmented | 20 |

## B.2 Test Results for Modeling Unique Instruments Without Segmentation



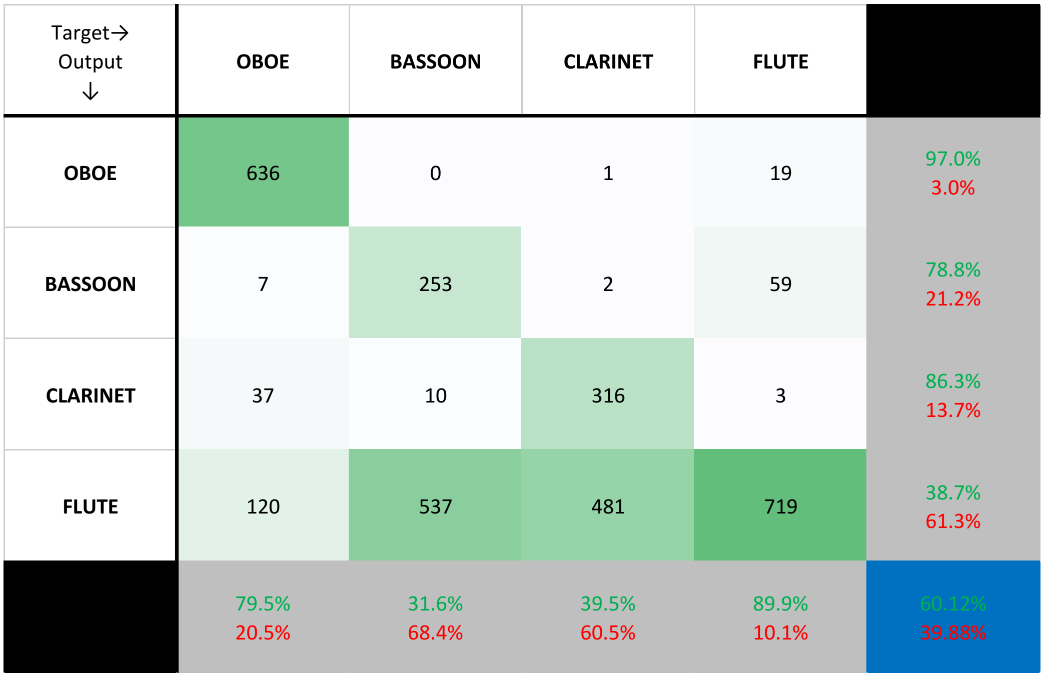
| Target→<br>Output<br>↓ | VIOLIN | VIOLA | CELLO | BASS | |
|---|---|---|---|---|---|
| VIOLIN | 500 | 103 | 34 | 52 | 72.6%<br>27.4% |
| VIOLA | 0 | 379 | 58 | 4 | 85.9%<br>14.1% |
| CELLO | 144 | 123 | 629 | 128 | 61.4%<br>38.6% |
| BASS | 156 | 195 | 79 | 616 | 58.9%<br>41.1% |
| | 62.5%<br>37.5% | 47.4%<br>52.6% | 78.6%<br>21.4% | 77.0%<br>77.8% | 66.38%<br>33.62% |

Figure B.1: Confusion Matrix from Strings Test using error-correcting output codes with support vector machines (ECOC) and naive bayes (NB) classifiers on CRBM Synthesis of 4096/2048 FFT



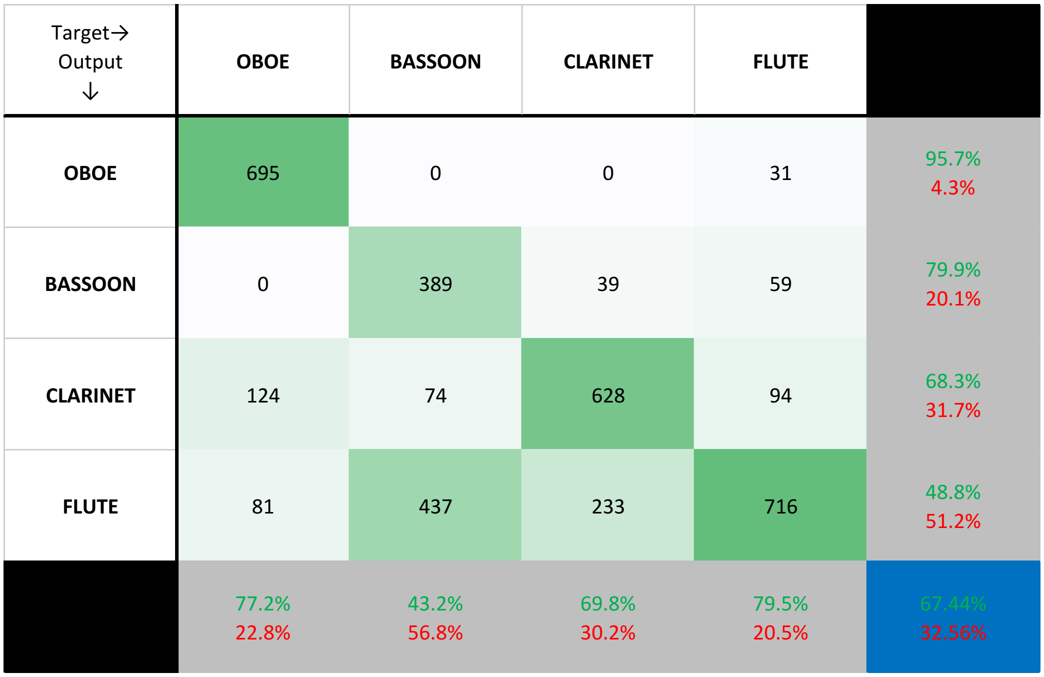
| Target→<br>Output<br>↓ | VIOLIN | VIOLA | CELLO | BASS | |
|---|---|---|---|---|---|
| VIOLIN | 615 | 39 | 63 | 66 | 78.5%<br>21.5% |
| VIOLA | 9 | 685 | 118 | 161 | 70.4%<br>29.6% |
| CELLO | 194 | 81 | 704 | 105 | 64.9%<br>35.1% |
| BASS | 82 | 95 | 15 | 568 | 74.7%<br>25.3% |
| | 68.3%<br>31.7% | 76.1%<br>23.9% | 78.2%<br>21.8% | 63.1%<br>36.9% | 71.44%<br>28.56% |

Figure B.2: Confusion Matrix from Strings Test using ECOC and NB on FCRBM Synthesis of 4096/2048 FFT



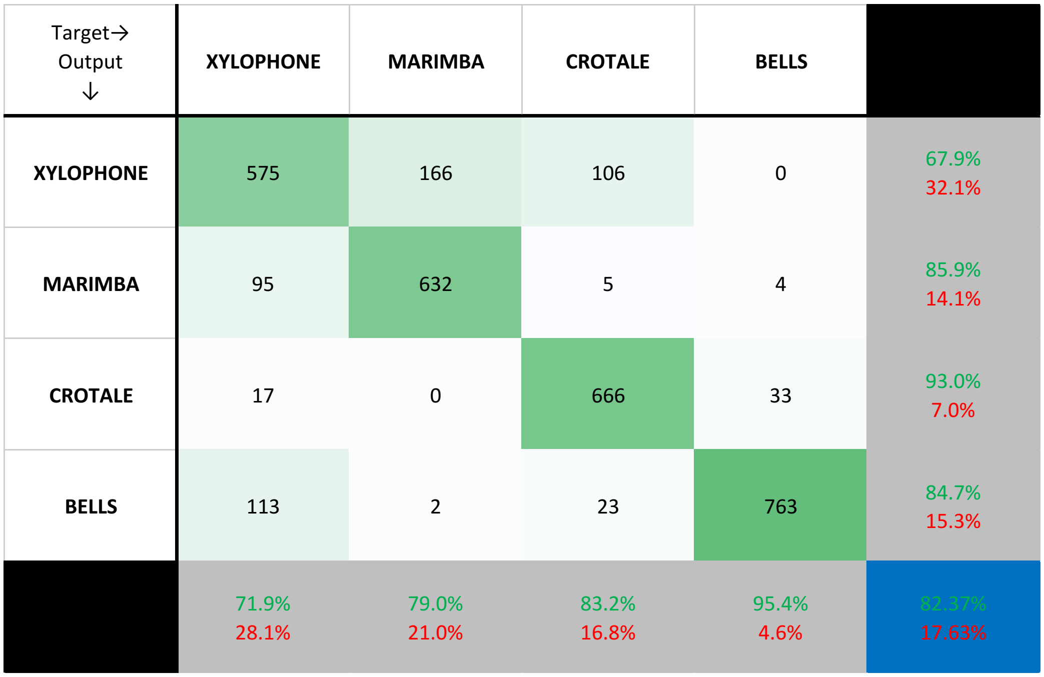
| Target→<br>Output<br>↓ | OBOE | BASSOON | CLARINET | FLUTE | |
|---|---|---|---|---|---|
| OBOE | 636 | 0 | 1 | 19 | 97.0%<br>3.0% |
| BASSOON | 7 | 253 | 2 | 59 | 78.8%<br>21.2% |
| CLARINET | 37 | 10 | 316 | 3 | 86.3%<br>13.7% |
| FLUTE | 120 | 537 | 481 | 719 | 38.7%<br>61.3% |
| | 79.5%<br>20.5% | 31.6%<br>68.4% | 39.5%<br>60.5% | 89.9%<br>10.1% | 60.12%<br>39.88% |

Figure B.3: Confusion Matrix from Woodwinds Test using ECOC and NB on CRBM Synthesis of 4096/2048



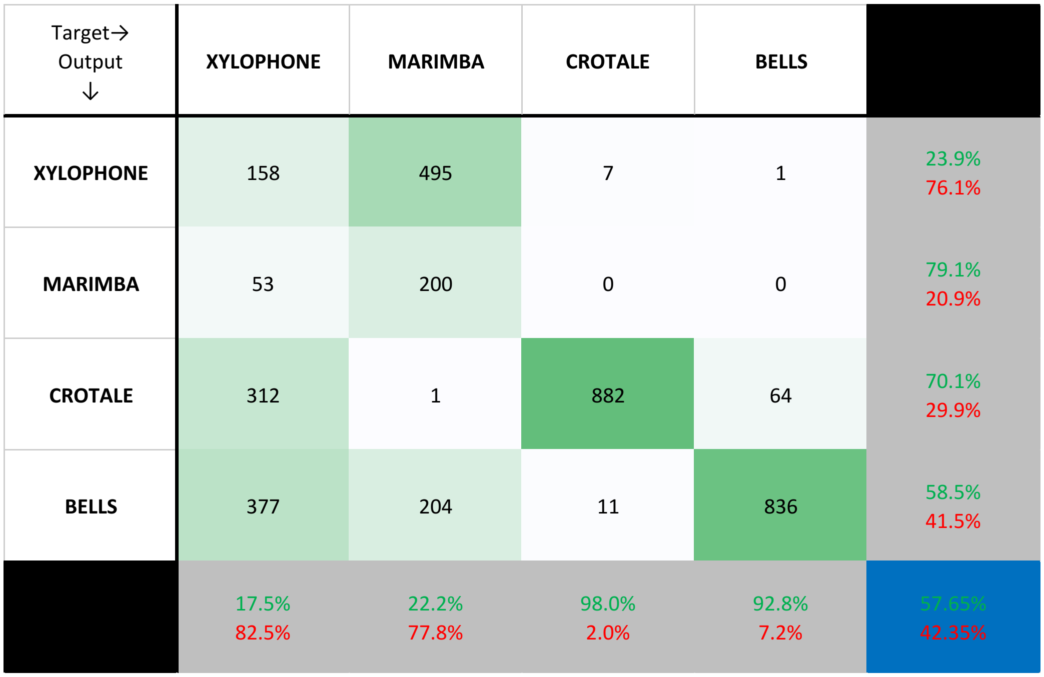
| Target→<br>Output<br>↓ | OBOE | BASSOON | CLARINET | FLUTE | |
|---|---|---|---|---|---|
| OBOE | 695 | 0 | 0 | 31 | 95.7%<br>4.3% |
| BASSOON | 0 | 389 | 39 | 59 | 79.9%<br>20.1% |
| CLARINET | 124 | 74 | 628 | 94 | 68.3%<br>31.7% |
| FLUTE | 81 | 437 | 233 | 716 | 48.8%<br>51.2% |
| | 77.2%<br>22.8% | 43.2%<br>56.8% | 69.8%<br>30.2% | 79.5%<br>20.5% | 67.44%<br>32.56% |

Figure B.4: Confusion Matrix from Woodwinds Test using ECOC and NB on FCRBM Synthesis of 4096/2048



| Target→<br>Output<br>↓ | XYLOPHONE | MARIMBA | CROTALE | BELLS | |
|---|---|---|---|---|---|
| XYLOPHONE | 575 | 166 | 106 | 0 | 67.9%<br>32.1% |
| MARIMBA | 95 | 632 | 5 | 4 | 85.9%<br>14.1% |
| CROTALE | 17 | 0 | 666 | 33 | 93.0%<br>7.0% |
| BELLS | 113 | 2 | 23 | 763 | 84.7%<br>15.3% |
| | 71.9%<br>28.1% | 79.0%<br>21.0% | 83.2%<br>16.8% | 95.4%<br>4.6% | 82.37%<br>17.63% |

Figure B.5: Confusion Matrix from Pitched Percussion Test using ECOC and NB on CRBM Synthesis of 4096/2048



| Target→<br>Output<br>↓ | XYLOPHONE | MARIMBA | CROTALE | BELLS | |
|---|---|---|---|---|---|
| XYLOPHONE | 158 | 495 | 7 | 1 | 23.9%<br>76.1% |
| MARIMBA | 53 | 200 | 0 | 0 | 79.1%<br>20.9% |
| CROTALE | 312 | 1 | 882 | 64 | 70.1%<br>29.9% |
| BELLS | 377 | 204 | 11 | 836 | 58.5%<br>41.5% |
| | 17.5%<br>82.5% | 22.2%<br>77.8% | 98.0%<br>2.0% | 92.8%<br>7.2% | 57.65%<br>42.35% |

Figure B.6: Confusion Matrix from Pitched Percussion Test using ECOC and NB on FCRBM Synthesis of 4096/2048

Table B.2: Resulting accuracies of ECOC/NB on tested models

| **MODEL** | STRINGS | WINDS | PERCUSSION |
|---|---|---|---|
| $C_{r1}$ | .95/.96 | .89/.69 | .90/.93 |
| $F_{r1}$ | .69/.59 | .61/.53 | .61/.71 |
| $C_{c1}$ | .74/.88 | .57/.54 | .87/.82 |
| $F_{c1}$ | .91/.85 | .80/.68 | .61/.55 |
| $C_{a1}$ | .25/.29 | .52/.40 | .63/.75 |
| $F_{a1}$ | .65/.66 | .63/.48 | .52/.52 |
| $C_{r5}$ | .91/.96 | .87/.75 | .96/.98 |
| $F_{r5}$ | .40/.20 | .72/.61 | .56/.47 |
| $C_{c5}$ | .62/.78 | .52/.32 | .90/.83 |
| $F_{c5}$ | .89/.85 | .86/.68 | .58/.55 |
| $C_{a5}$ | .18/.41 | .52/.33 | .56/.73 |
| $F_{a5}$ | .79/.66 | .69/.52 | .57/.53 |
| $C_{r20}$ | .93/.96 | .96/.79 | .73/.97 |
| $F_{r20}$ | .53/.67 | .79/.83 | .63/.56 |
| $C_{c20}$ | .79/.72 | .96/.37 | .84/.96 |
| $F_{c20}$ | .88/.84 | .85/.67 | .63/.67 |
| $C_{a20}$ | .24/.53 | .56/.31 | .77/.82 |
| $F_{a20}$ | .82/.83 | .67/.49 | .56/.52 |

Table B.3: Resulting accuracies of ECOC/NB on $C_{model}$

| **MODEL** | STRINGS | WINDS | PERCUSSION |
|---|---|---|---|
| $C_{r1}$ | .95/.96 | .89/.86 | .90/.93 |
| $C_{c1}$ | .74/.88 | .57/.54 | .87/.82 |
| $C_{a1}$ | .25/.29 | .52/.40 | .63/.75 |
| $C_{r5}$ | .91/.96 | .87/.75 | .96/.98 |
| $C_{c5}$ | .62/.78 | .52/.32 | .90/.83 |
| $C_{a5}$ | .18/.41 | .52/.33 | .56/.73 |
| $C_{r20}$ | .93/.96 | .96/.79 | .73/.97 |
| $C_{c20}$ | .79/.72 | .96/.37 | .84/.96 |
| $C_{a20}$ | .24/.53 | .56/.31 | .77/.82 |

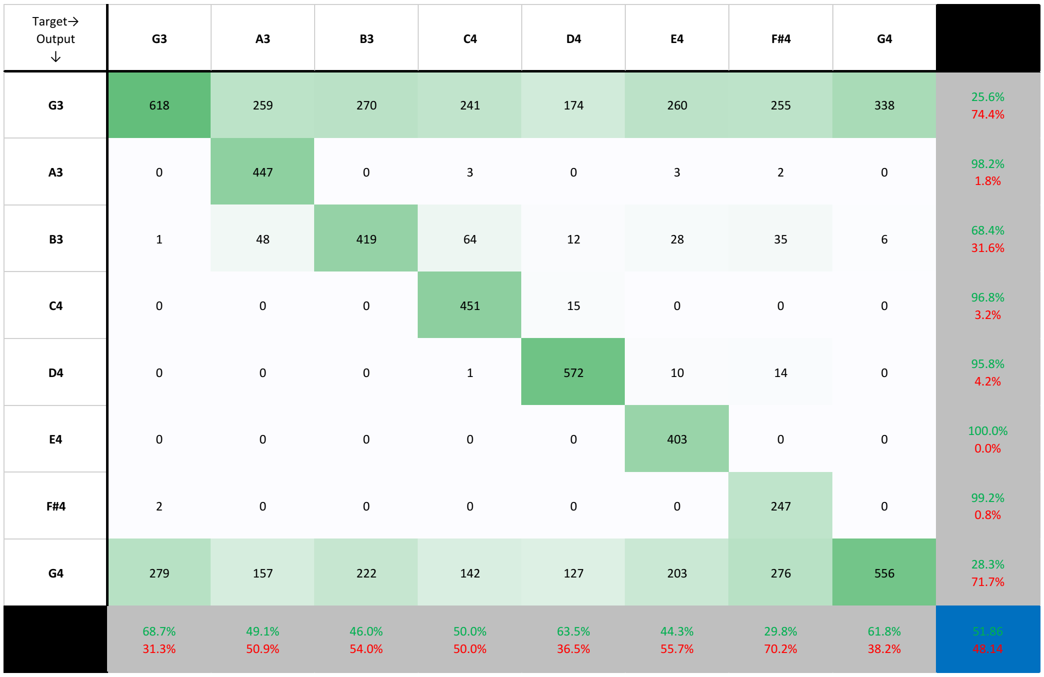
Table B.4: Resulting accuracies of ECOC on $F_{model}$

| **MODEL** | STRINGS | WINDS | PERCUSSION |
|---|---|---|---|
| $F_{r1}$ | .69/.59 | .61/.53 | .61/.71 |
| $F_{c1}$ | .91/.85 | .80/.68 | .61/.55 |
| $F_{a1}$ | .65/.66 | .63/.48 | .52/.52 |
| $F_{r5}$ | .40/.20 | .72/.61 | .56/.47 |
| $F_{c5}$ | .89/.85 | .86/.68 | .58/.55 |
| $F_{a5}$ | .79/.66 | .69/.52 | .57/.53 |
| $F_{r20}$ | .53/.67 | .79/.83 | .63/.56 |
| $F_{c20}$ | .88/.84 | .85/.67 | .63/.67 |
| $F_{a20}$ | .82/.83 | .67/.49 | .56/.52 |

Table B.5: Aggregate average accuracies of SVM/NB classifiers on models and parameters

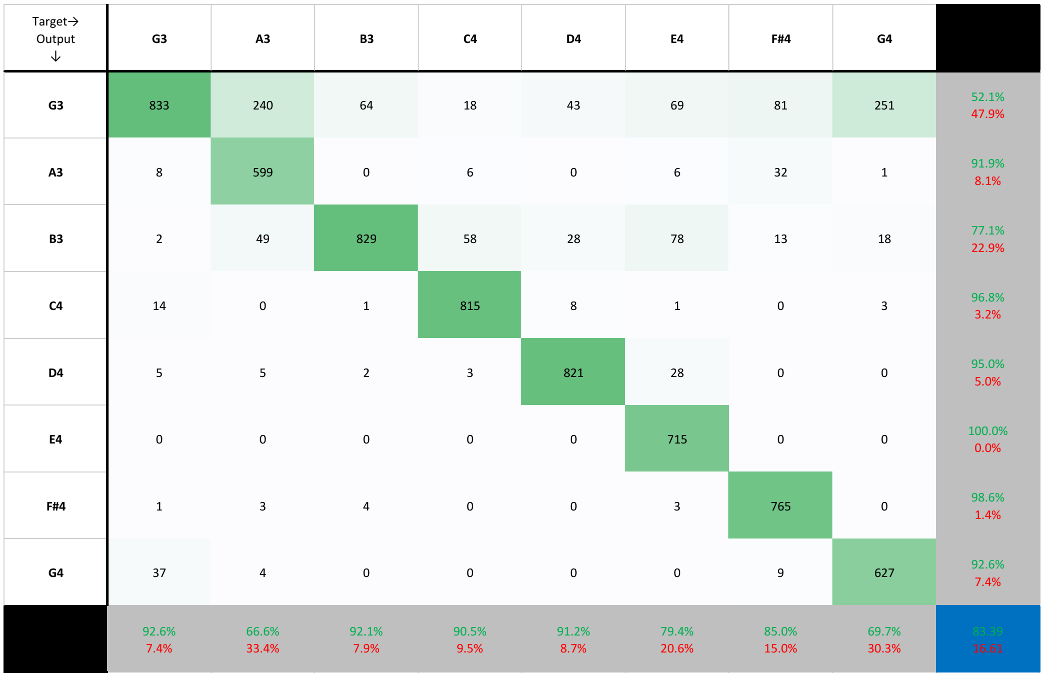
| **Model** | Accuracy | **NumPars** | Accuracy | **CD/Gibbs** | Accuracy |
|---|---|---|---|---|---|
| $C$ | .709/695 | $r$ | .758/.731 | 1 | .686/.657 |
| $F$ | .683/619 | $c$ | .768/.700 | 5 | .672/.620 |
| | | $a$ | .533/.563 | 20 | .730/.695 |

## B.3 Test Results for Modeling Pitches of Individual Instruments Without Segmentation



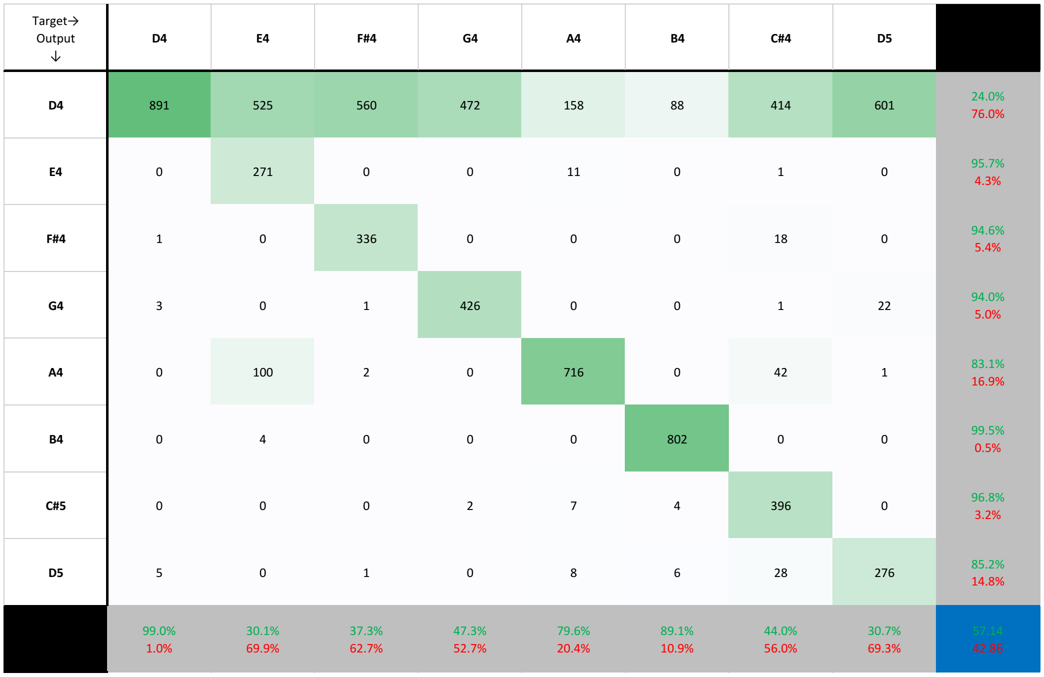
| Target→<br>Output<br>↓ | G3 | A3 | B3 | C4 | D4 | E4 | F#4 | G4 | |
|---|---|---|---|---|---|---|---|---|---|
| G3 | 618 | 259 | 270 | 241 | 174 | 260 | 255 | 338 | 25.6%<br>74.4% |
| A3 | 0 | 447 | 0 | 3 | 0 | 3 | 2 | 0 | 98.2%<br>1.8% |
| B3 | 1 | 48 | 419 | 64 | 12 | 28 | 35 | 6 | 68.4%<br>31.6% |
| C4 | 0 | 0 | 0 | 451 | 15 | 0 | 0 | 0 | 96.8%<br>3.2% |
| D4 | 0 | 0 | 0 | 1 | 572 | 10 | 14 | 0 | 95.8%<br>4.2% |
| E4 | 0 | 0 | 0 | 0 | 0 | 403 | 0 | 0 | 100.0%<br>0.0% |
| F#4 | 2 | 0 | 0 | 0 | 0 | 0 | 247 | 0 | 99.2%<br>0.8% |
| G4 | 279 | 157 | 222 | 142 | 127 | 203 | 276 | 556 | 28.3%<br>71.7% |
| | 68.7%<br>31.3% | 49.1%<br>50.9% | 46.0%<br>54.0% | 50.0%<br>50.0% | 63.5%<br>36.5% | 44.3%<br>55.7% | 29.8%<br>70.2% | 61.8%<br>38.2% | 51.86<br>48.14 |

Figure B.7: Confusion Matrix from Violin Pitches Test using ECOC and NB on CRBM Synthesis of 4096/2048 FFT



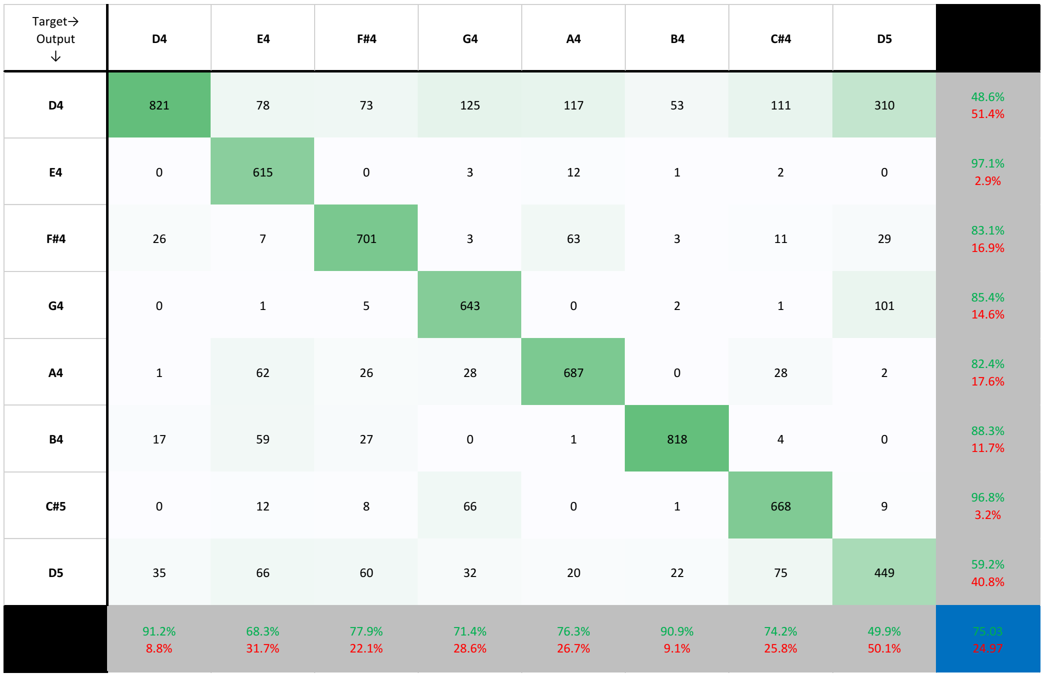
| Target→<br>Output<br>↓ | G3 | A3 | B3 | C4 | D4 | E4 | F#4 | G4 | |
|---|---|---|---|---|---|---|---|---|---|
| G3 | 833 | 240 | 64 | 18 | 43 | 69 | 81 | 251 | 52.1%<br>47.9% |
| A3 | 8 | 599 | 0 | 6 | 0 | 6 | 32 | 1 | 91.9%<br>8.1% |
| B3 | 2 | 49 | 829 | 58 | 28 | 78 | 13 | 18 | 77.1%<br>22.9% |
| C4 | 14 | 0 | 1 | 815 | 8 | 1 | 0 | 3 | 96.8%<br>3.2% |
| D4 | 5 | 5 | 2 | 3 | 821 | 28 | 0 | 0 | 95.0%<br>5.0% |
| E4 | 0 | 0 | 0 | 0 | 0 | 715 | 0 | 0 | 100.0%<br>0.0% |
| F#4 | 1 | 3 | 4 | 0 | 0 | 3 | 765 | 0 | 98.6%<br>1.4% |
| G4 | 37 | 4 | 0 | 0 | 0 | 0 | 9 | 627 | 92.6%<br>7.4% |
| | 92.6%<br>7.4% | 66.6%<br>33.4% | 92.1%<br>7.9% | 90.5%<br>9.5% | 91.2%<br>8.7% | 79.4%<br>20.6% | 85.0%<br>15.0% | 69.7%<br>30.3% | 83.39<br>16.61 |

Figure B.8: Confusion Matrix from Violin Pitches Test using ECOC and NB on FCRBM Synthesis of 4096/2048 FFT



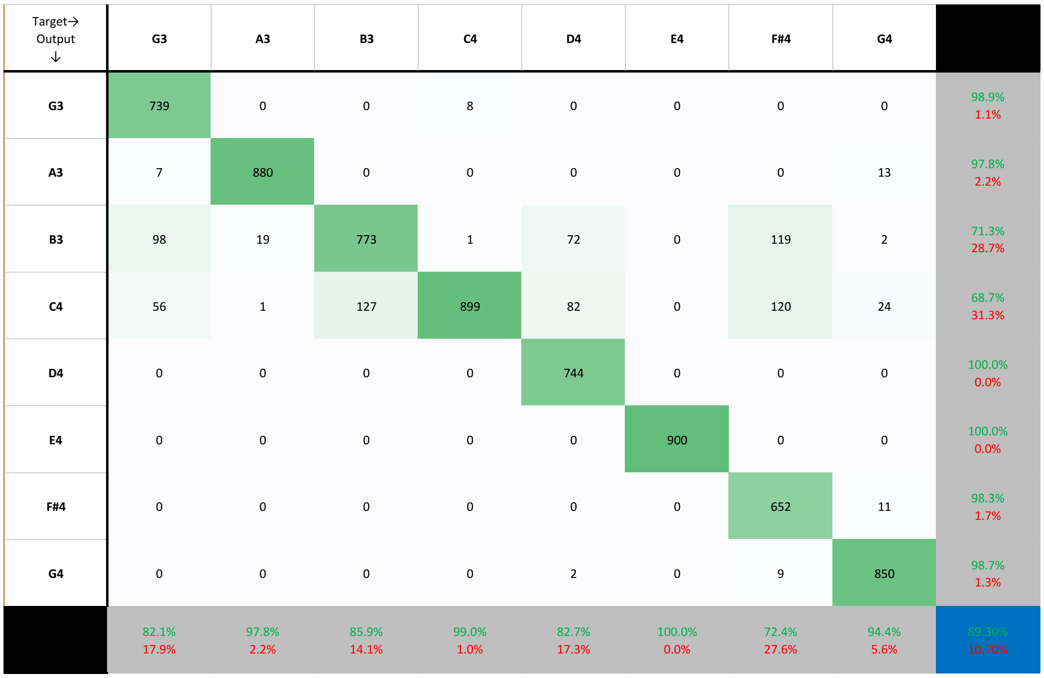
| Target→<br>Output<br>↓ | D4 | E4 | F#4 | G4 | A4 | B4 | C#4 | D5 | |
|---|---|---|---|---|---|---|---|---|---|
| D4 | 891 | 525 | 560 | 472 | 158 | 88 | 414 | 601 | 24.0%<br>76.0% |
| E4 | 0 | 271 | 0 | 0 | 11 | 0 | 1 | 0 | 95.7%<br>4.3% |
| F#4 | 1 | 0 | 336 | 0 | 0 | 0 | 18 | 0 | 94.6%<br>5.4% |
| G4 | 3 | 0 | 1 | 426 | 0 | 0 | 1 | 22 | 94.0%<br>5.0% |
| A4 | 0 | 100 | 2 | 0 | 716 | 0 | 42 | 1 | 83.1%<br>16.9% |
| B4 | 0 | 4 | 0 | 0 | 0 | 802 | 0 | 0 | 99.5%<br>0.5% |
| C#5 | 0 | 0 | 0 | 2 | 7 | 4 | 396 | 0 | 96.8%<br>3.2% |
| D5 | 5 | 0 | 1 | 0 | 8 | 6 | 28 | 276 | 85.2%<br>14.8% |
| | 99.0%<br>1.0% | 30.1%<br>69.9% | 37.3%<br>62.7% | 47.3%<br>52.7% | 79.6%<br>20.4% | 89.1%<br>10.9% | 44.0%<br>56.0% | 30.7%<br>69.3% | 57.14<br>42.86 |

Figure B.9: Confusion Matrix from Oboe Pitches Test using ECOC and NB on CRBM Synthesis of 4096/2048 FFT



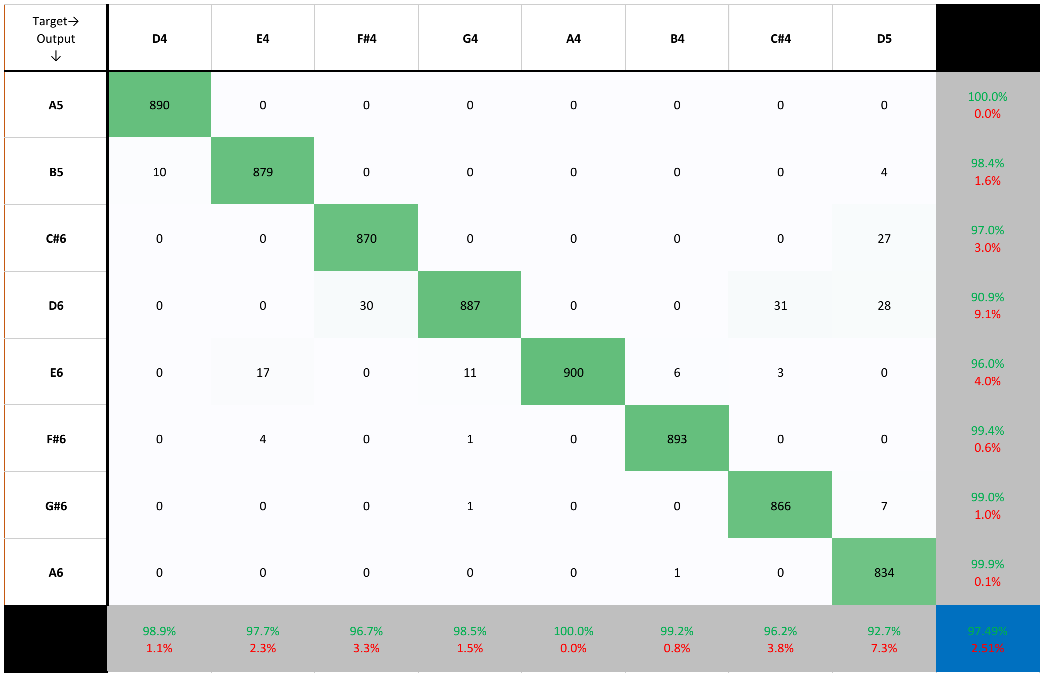
| Target→<br>Output<br>↓ | D4 | E4 | F#4 | G4 | A4 | B4 | C#4 | D5 | |
|---|---|---|---|---|---|---|---|---|---|
| D4 | 821 | 78 | 73 | 125 | 117 | 53 | 111 | 310 | 48.6%<br>51.4% |
| E4 | 0 | 615 | 0 | 3 | 12 | 1 | 2 | 0 | 97.1%<br>2.9% |
| F#4 | 26 | 7 | 701 | 3 | 63 | 3 | 11 | 29 | 83.1%<br>16.9% |
| G4 | 0 | 1 | 5 | 643 | 0 | 2 | 1 | 101 | 85.4%<br>14.6% |
| A4 | 1 | 62 | 26 | 28 | 687 | 0 | 28 | 2 | 82.4%<br>17.6% |
| B4 | 17 | 59 | 27 | 0 | 1 | 818 | 4 | 0 | 88.3%<br>11.7% |
| C#5 | 0 | 12 | 8 | 66 | 0 | 1 | 668 | 9 | 96.8%<br>3.2% |
| D5 | 35 | 66 | 60 | 32 | 20 | 22 | 75 | 449 | 59.2%<br>40.8% |
| | 91.2%<br>8.8% | 68.3%<br>31.7% | 77.9%<br>22.1% | 71.4%<br>28.6% | 76.3%<br>26.7% | 90.9%<br>9.1% | 74.2%<br>25.8% | 49.9%<br>50.1% | 75.03<br>24.97 |

Figure B.10: Confusion Matrix from Oboe Pitches Test using ECOC and NB on FCRBM Synthesis of 4096/2048 FFT



| Target→<br>Output<br>↓ | G3 | A3 | B3 | C4 | D4 | E4 | F#4 | G4 | |
|---|---|---|---|---|---|---|---|---|---|
| G3 | 739 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 98.9%<br>1.1% |
| A3 | 7 | 880 | 0 | 0 | 0 | 0 | 0 | 13 | 97.8%<br>2.2% |
| B3 | 98 | 19 | 773 | 1 | 72 | 0 | 119 | 2 | 71.3%<br>28.7% |
| C4 | 56 | 1 | 127 | 899 | 82 | 0 | 120 | 24 | 68.7%<br>31.3% |
| D4 | 0 | 0 | 0 | 0 | 744 | 0 | 0 | 0 | 100.0%<br>0.0% |
| E4 | 0 | 0 | 0 | 0 | 0 | 900 | 0 | 0 | 100.0%<br>0.0% |
| F#4 | 0 | 0 | 0 | 0 | 0 | 0 | 652 | 11 | 98.3%<br>1.7% |
| G4 | 0 | 0 | 0 | 0 | 2 | 0 | 9 | 850 | 98.7%<br>1.3% |
| | 82.1%<br>17.9% | 97.8%<br>2.2% | 85.9%<br>14.1% | 99.0%<br>1.0% | 82.7%<br>17.3% | 100.0%<br>0.0% | 72.4%<br>27.6% | 94.4%<br>5.6% | 89.30%<br>10.70% |

Figure B.11: Confusion Matrix from Bells Pitches Test using ECOC and NB on CRBM Synthesis of 4096/2048 FFT



| Target→<br>Output<br>↓ | D4 | E4 | F#4 | G4 | A4 | B4 | C#4 | D5 | |
|---|---|---|---|---|---|---|---|---|---|
| A5 | 890 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100.0%<br>0.0% |
| B5 | 10 | 879 | 0 | 0 | 0 | 0 | 0 | 4 | 98.4%<br>1.6% |
| C#6 | 0 | 0 | 870 | 0 | 0 | 0 | 0 | 27 | 97.0%<br>3.0% |
| D6 | 0 | 0 | 30 | 887 | 0 | 0 | 31 | 28 | 90.9%<br>9.1% |
| E6 | 0 | 17 | 0 | 11 | 900 | 6 | 3 | 0 | 96.0%<br>4.0% |
| F#6 | 0 | 4 | 0 | 1 | 0 | 893 | 0 | 0 | 99.4%<br>0.6% |
| G#6 | 0 | 0 | 0 | 1 | 0 | 0 | 866 | 7 | 99.0%<br>1.0% |
| A6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 834 | 99.9%<br>0.1% |
| | 98.9%<br>1.1% | 97.7%<br>2.3% | 96.7%<br>3.3% | 98.5%<br>1.5% | 100.0%<br>0.0% | 99.2%<br>0.8% | 96.2%<br>3.8% | 92.7%<br>7.3% | 97.49%<br>2.51% |

Figure B.12: Confusion Matrix from Bells Pitches Test using ECOC and NB on FCRBM Synthesis of 4096/2048 FFT

Table B.6: Resulting accuracies of ECOC/NB on tested models

| **MODEL** | VIOLIN SCALE | OBOE SCALE | BELLS SCALE |
|---|---|---|---|
| $C_{r1}$ | .98/.79 | .92/.97 | 1.00/1.00 |
| $F_{r1}$ | .54/.78 | .64/.85 | .88/.90 |
| $C_{c1}$ | .59/.52 | .48/.33 | 1.00/1.00 |
| $F_{c1}$ | .97/.72 | .93/.94 | 1.00/.99 |
| $C_{a1}$ | .23/.24 | .34/.22 | .93/.91 |
| $F_{a1}$ | .96/.89 | .72/.95 | 1.00/.99 |
| | | | |
| $C_{r5}$ | .96/.67 | .92/.95 | 1.00/1.00 |
| $F_{r5}$ | .81/.94 | .50/.47 | .96/.91 |
| $C_{c5}$ | .60/.45 | .44/.43 | .75/.62 |
| $F_{c5}$ | .97/.76 | .96/.92 | 1.00/.98 |
| $C_{a5}$ | .20/.19 | .42/.32 | .78/.72 |
| $F_{a5}$ | .82/.72 | .92/.93 | 1.00/.99 |
| | | | |
| $C_{r20}$ | .96/.67 | .94/.97 | 1.00/1.00 |
| $F_{r20}$ | .81/.95 | .57/.87 | .99/.94 |
| $C_{c20}$ | .49/.38 | .50/.44 | .91/.76 |
| $F_{c20}$ | .96/.76 | .94/.93 | 1.00/.99 |
| $C_{a20}$ | .22/.20 | .40/.27 | .92/.78 |
| $F_{a20}$ | .91/.82 | .83/.92 | 1.00/.99 |

Table B.7: Resulting accuracies of ECOC/NB on $C_{model}$

| **MODEL** | VIOLIN SCALE | OBOE SCALE | BELLS SCALE |
|---|---|---|---|
| $C_{r1}$ | .98/.79 | .92/.97 | 1.00/1.00 |
| $C_{c1}$ | .59/.52 | .48/.33 | 1.00/1.00 |
| $C_{a1}$ | .23/.24 | .34/.22 | .93/.91 |
| $C_{r5}$ | .96/.67 | .92/.95 | 1.00/1.00 |
| $C_{c5}$ | .60/.45 | .44/.43 | .75/.62 |
| $C_{a5}$ | .20/.19 | .42/.32 | .78/.72 |
| $C_{r20}$ | .96/.67 | .94/.97 | 1.00/1.00 |
| $C_{c20}$ | .49/.38 | .50/.44 | .91/.76 |
| $C_{a20}$ | .22/.20 | .40/.27 | .92/.78 |

Table B.8: Resulting accuracies of ECOC/NB on $F_{model}$

| **MODEL** | VIOLIN SCALE | OBOE SCALE | BELLS SCALE |
|---|---|---|---|
| $F_{r1}$ | .54/.78 | .64/.85 | .88/.90 |
| $F_{c1}$ | .97/.72 | .93/.94 | 1.00/.99 |
| $F_{a1}$ | .96/.89 | .72/.95 | 1.00/.99 |
| $F_{r5}$ | .81/.94 | .50/.47 | .96/.91 |
| $F_{c5}$ | .97/.76 | .96/.92 | 1.00/.98 |
| $F_{a5}$ | .82/.72 | .92/.93 | 1.00/.99 |
| $F_{r20}$ | .81/.95 | .57/.87 | .99/.94 |
| $F_{c20}$ | .96/.76 | .94/.93 | 1.00/.99 |
| $F_{a20}$ | .91/.82 | .83/.92 | 1.00/.99 |

Table B.9: Aggregate average accuracies of SVM/NB classifiers on models and parameters

| **Model** | Accuracy | **NumPars** | Accuracy | **CD/Gibbs** | Accuracy |
|---|---|---|---|---|---|
| $C$ | .700/.622 | $r$ | .854/.868 | 1 | .782/.777 |
| $F$ | .873/.881 | $c$ | .805/.718 | 5 | .778/.721 |
|  |  | $a$ | .663/.699 | 20 | .798/.758 |

# APPENDIX C

# DEEP BELIEF NET EXPERIMENTS

## C.1 An Implementation of a Deep Factored Conditional Restricted Boltzmann Machine (dFCRBM)

A dFCRBM learns connections across layers of the algorithms, in this case, a given number of FCRBM, using the learned hidden representation at each layer as input to the next layer, creating a deep belief net.

Each layer of the deep belief net is learned in steps, generating outputs for each layer and using those outputs to learn deeper layer connections. For example, the first layer of connected hidden units is learned the same as a in a single layer FCRBM. The resulting connections are then used to generate the hidden units as an output (similar to the role visible units take in the first layer) to learn the connections to the second layer of hidden units, where previous timesteps of the first layer of hidden units are used for the autoregressive connections.

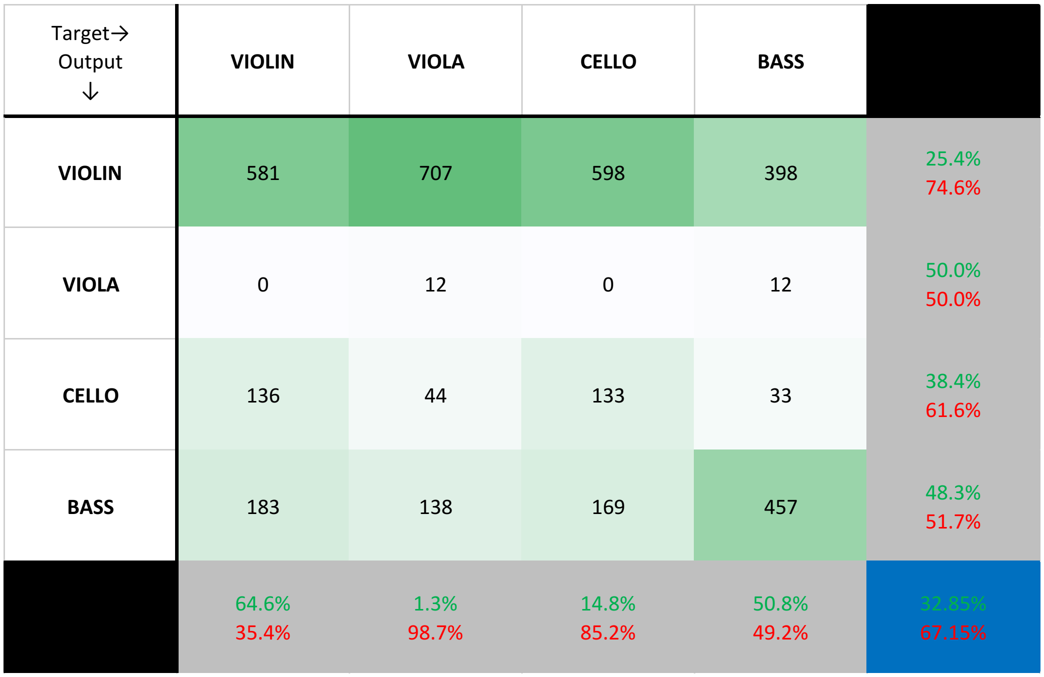
The equations for the dFCRBM are the same as those that are used for its composite layers. For example, if the dFCRBM was a 2-layered net, with the first layer consisting of a FCRBM and the second layer consisting of a CRBM, the dFCRBM would first learn the necessary weights of the FCRBM as done in 2.2.4 and then generate hidden units from equation 2.28 using these learned weights. Those hidden unit outputs would then be used as visible/sample data for the second layer CRBM to learn, using the same equations in 2.2.3. Data is synthesized by doing a reconstructive pass forward, through both layers of the dFCRBM, using the first layer's output to drive the second layer's synthesis of the data.

## C.2 List of Tested Deep Models

Table C.1: Names and parameters of tested models

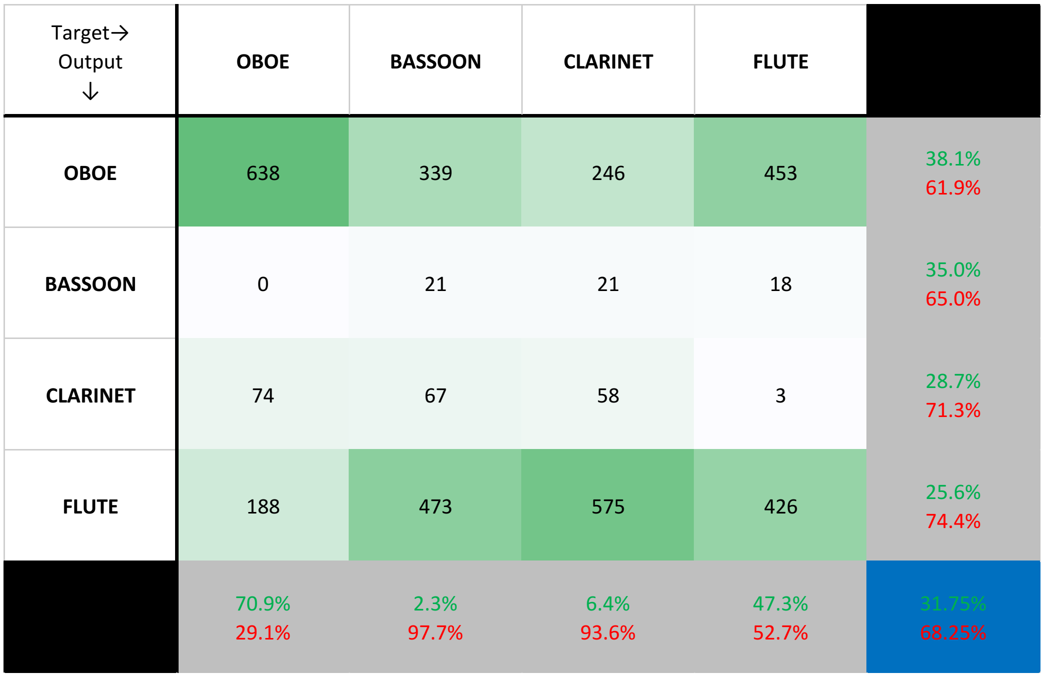
| MODEL | ALGORITHM | INTERNAL PARAMETERS | ITERATIONS |
|---|---|---|---|
| | | (numHid, numFac, numFeat) | (numCD, numGibbs |
| $D_{r1}$ | $dFCRBM$ | redDim | 1 |
| $D_{c1}$ | $dFCRBM$ | constant | 1 |
| $D_{a1}$ | $dFCRBM$ | augmented | 1 |
| $D_{r5}$ | $dFCRBM$ | redDim | 5 |
| $D_{c5}$ | $dFCRBM$ | constant | 5 |
| $D_{a5}$ | $dFCRBM$ | augmented | 5 |
| $D_{r20}$ | $dFCRBM$ | redDim | 20 |
| $D_{c20}$ | $dFCRBM$ | constant | 20 |
| $D_{a20}$ | $dFCRBM$ | augmented | 20 |

## C.3 Test Results for Modeling Unique Instruments Without Segmentation



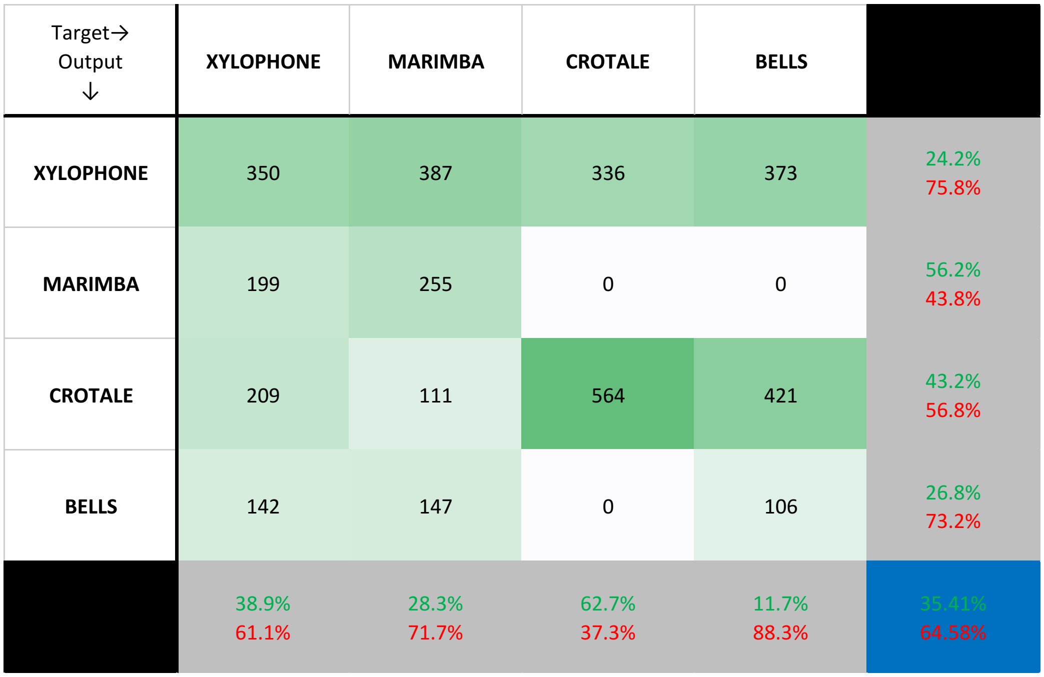
| Target→<br>Output<br>↓ | VIOLIN | VIOLA | CELLO | BASS | |
|---|---|---|---|---|---|
| VIOLIN | 581 | 707 | 598 | 398 | 25.4%<br>74.6% |
| VIOLA | 0 | 12 | 0 | 12 | 50.0%<br>50.0% |
| CELLO | 136 | 44 | 133 | 33 | 38.4%<br>61.6% |
| BASS | 183 | 138 | 169 | 457 | 48.3%<br>51.7% |
| | 64.6%<br>35.4% | 1.3%<br>98.7% | 14.8%<br>85.2% | 50.8%<br>49.2% | 32.85%<br>67.15% |

Figure C.1: Confusion Matrix from Strings Test using ECOC and NB on dFCRBM Synthesis of 4096/2048



| Target→<br>Output<br>↓ | OBOE | BASSOON | CLARINET | FLUTE | |
|---|---|---|---|---|---|
| OBOE | 638 | 339 | 246 | 453 | 38.1%<br>61.9% |
| BASSOON | 0 | 21 | 21 | 18 | 35.0%<br>65.0% |
| CLARINET | 74 | 67 | 58 | 3 | 28.7%<br>71.3% |
| FLUTE | 188 | 473 | 575 | 426 | 25.6%<br>74.4% |
| | 70.9%<br>29.1% | 2.3%<br>97.7% | 6.4%<br>93.6% | 47.3%<br>52.7% | 31.75%<br>68.25% |

Figure C.2: Confusion Matrix from Woodwinds Test using ECOC and NB on dFCRBM Synthesis of 4096/2048



| Target→<br>Output<br>↓ | XYLOPHONE | MARIMBA | CROTALE | BELLS | |
|---|---|---|---|---|---|
| XYLOPHONE | 350 | 387 | 336 | 373 | 24.2%<br>75.8% |
| MARIMBA | 199 | 255 | 0 | 0 | 56.2%<br>43.8% |
| CROTALE | 209 | 111 | 564 | 421 | 43.2%<br>56.8% |
| BELLS | 142 | 147 | 0 | 106 | 26.8%<br>73.2% |
| | 38.9%<br>61.1% | 28.3%<br>71.7% | 62.7%<br>37.3% | 11.7%<br>88.3% | 35.41%<br>64.58% |

Figure C.3: Confusion Matrix from Pitched Percussion Test using ECOC and NB on dFCRBM Synthesis of 4096/2048

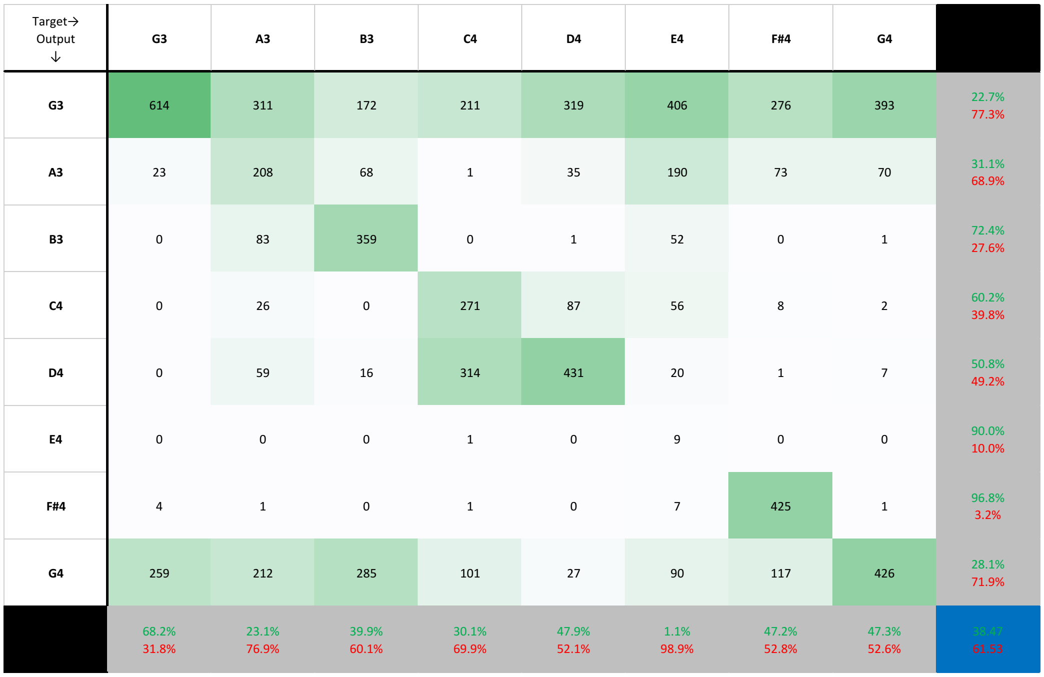
Table C.2: Resulting accuracies of ECOC/NB on tested models

| **MODEL** | STRINGS | WINDS | PERCUSSION |
|---|---|---|---|
| $D_{r1}$ | .46/.25 | .30/.28 | .32/.25 |
| $D_{c1}$ | .42/.28 | .31/.24 | .38/.48 |
| $D_{a1}$ | .47/.31 | .38/.33 | .61/.22 |
| | | | |
| $D_{r5}$ | .35/.19 | .42/.28 | .53/.39 |
| $D_{c5}$ | .29/.30 | .39/.19 | .46/.05 |
| $D_{a5}$ | .17/.30 | .35/.32 | .53/.04 |
| | | | |
| $D_{r20}$ | .37/.22 | .33/.31 | .38/.18 |
| $D_{c20}$ | .50/.30 | .27/.39 | .42/.08 |
| $D_{a20}$ | .51/.19 | .29/.29 | .63/.41 |

Table C.3: Resulting accuracies of ECOC on $D_{model}$

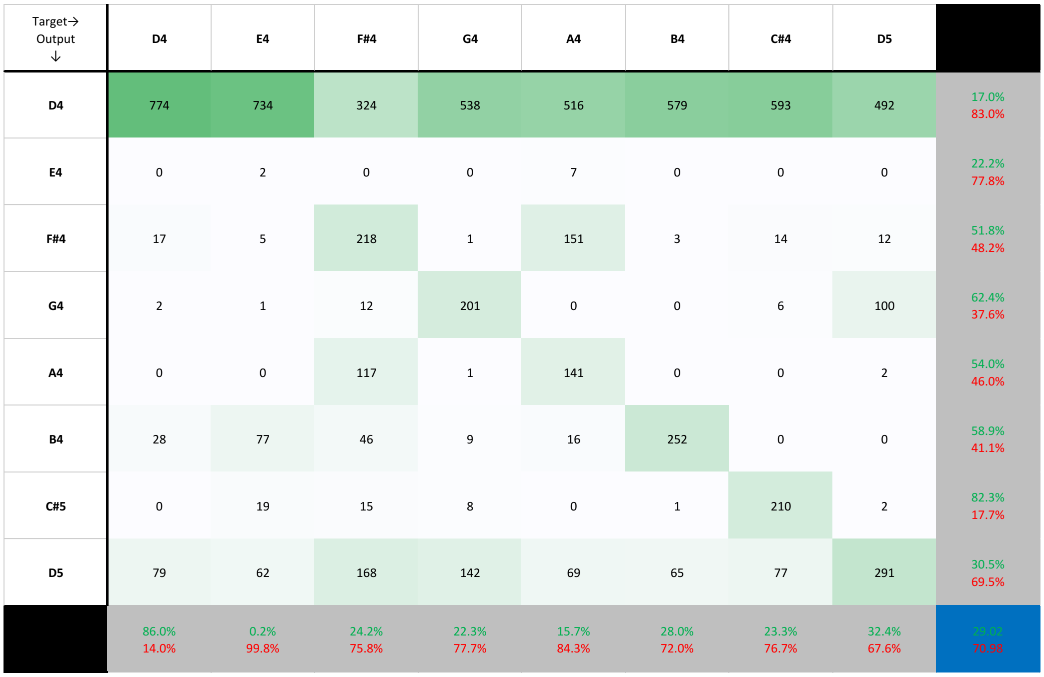
| **MODEL** | STRINGS | WINDS | PERCUSSION |
|---|---|---|---|
| $D_{r1}$ | .46/.25 | .30/.28 | .32/.25 |
| $D_{c1}$ | .42/.28 | .31/.24 | .38/.48 |
| $D_{a1}$ | .47/.31 | .38/.33 | .61/.22 |
| $D_{r5}$ | .35/.19 | .42/.28 | .53/.39 |
| $D_{c5}$ | .29/.30 | .39/.19 | .46/.05 |
| $D_{a5}$ | .17/.30 | .35/.32 | .53/.04 |
| $D_{r20}$ | .37/.22 | .33/.31 | .38/.18 |
| $D_{c20}$ | .50/.30 | .27/.39 | .42/.08 |
| $D_{a20}$ | .51/.19 | .29/.29 | .63/.41 |

## C.4 Test Results for Modeling Pitches of Individual Instruments Without Segmentation



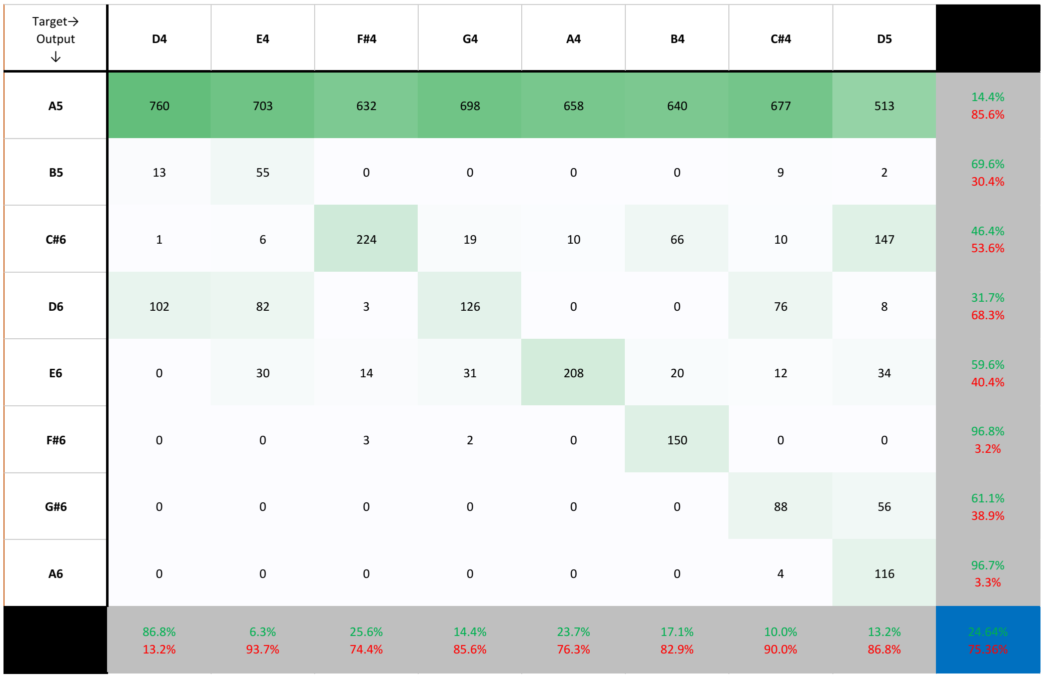
| Target→<br>Output<br>↓ | G3 | A3 | B3 | C4 | D4 | E4 | F#4 | G4 | |
|---|---|---|---|---|---|---|---|---|---|
| G3 | 614 | 311 | 172 | 211 | 319 | 406 | 276 | 393 | 22.7%<br>77.3% |
| A3 | 23 | 208 | 68 | 1 | 35 | 190 | 73 | 70 | 31.1%<br>68.9% |
| B3 | 0 | 83 | 359 | 0 | 1 | 52 | 0 | 1 | 72.4%<br>27.6% |
| C4 | 0 | 26 | 0 | 271 | 87 | 56 | 8 | 2 | 60.2%<br>39.8% |
| D4 | 0 | 59 | 16 | 314 | 431 | 20 | 1 | 7 | 50.8%<br>49.2% |
| E4 | 0 | 0 | 0 | 1 | 0 | 9 | 0 | 0 | 90.0%<br>10.0% |
| F#4 | 4 | 1 | 0 | 1 | 0 | 7 | 425 | 1 | 96.8%<br>3.2% |
| G4 | 259 | 212 | 285 | 101 | 27 | 90 | 117 | 426 | 28.1%<br>71.9% |
| | 68.2%<br>31.8% | 23.1%<br>76.9% | 39.9%<br>60.1% | 30.1%<br>69.9% | 47.9%<br>52.1% | 1.1%<br>98.9% | 47.2%<br>52.8% | 47.3%<br>52.6% | 38.47<br>61.53 |

Figure C.4: Confusion Matrix from Violin Pitches Test using ECOC and NB on dFCRBM Synthesis of 4096/2048 FFT



| Target→<br>Output<br>↓ | D4 | E4 | F#4 | G4 | A4 | B4 | C#4 | D5 | |
|---|---|---|---|---|---|---|---|---|---|
| D4 | 774 | 734 | 324 | 538 | 516 | 579 | 593 | 492 | 17.0%<br>83.0% |
| E4 | 0 | 2 | 0 | 0 | 7 | 0 | 0 | 0 | 22.2%<br>77.8% |
| F#4 | 17 | 5 | 218 | 1 | 151 | 3 | 14 | 12 | 51.8%<br>48.2% |
| G4 | 2 | 1 | 12 | 201 | 0 | 0 | 6 | 100 | 62.4%<br>37.6% |
| A4 | 0 | 0 | 117 | 1 | 141 | 0 | 0 | 2 | 54.0%<br>46.0% |
| B4 | 28 | 77 | 46 | 9 | 16 | 252 | 0 | 0 | 58.9%<br>41.1% |
| C#5 | 0 | 19 | 15 | 8 | 0 | 1 | 210 | 2 | 82.3%<br>17.7% |
| D5 | 79 | 62 | 168 | 142 | 69 | 65 | 77 | 291 | 30.5%<br>69.5% |
| | 86.0%<br>14.0% | 0.2%<br>99.8% | 24.2%<br>75.8% | 22.3%<br>77.7% | 15.7%<br>84.3% | 28.0%<br>72.0% | 23.3%<br>76.7% | 32.4%<br>67.6% | 29.02<br>70.98 |

Figure C.5: Confusion Matrix from Oboe Pitches Test using ECOC and NB on dFCRBM Synthesis of 4096/2048 FFT



| Target→<br>Output<br>↓ | D4 | E4 | F#4 | G4 | A4 | B4 | C#4 | D5 | |
|---|---|---|---|---|---|---|---|---|---|
| A5 | 760 | 703 | 632 | 698 | 658 | 640 | 677 | 513 | 14.4%<br>85.6% |
| B5 | 13 | 55 | 0 | 0 | 0 | 0 | 9 | 2 | 69.6%<br>30.4% |
| C#6 | 1 | 6 | 224 | 19 | 10 | 66 | 10 | 147 | 46.4%<br>53.6% |
| D6 | 102 | 82 | 3 | 126 | 0 | 0 | 76 | 8 | 31.7%<br>68.3% |
| E6 | 0 | 30 | 14 | 31 | 208 | 20 | 12 | 34 | 59.6%<br>40.4% |
| F#6 | 0 | 0 | 3 | 2 | 0 | 150 | 0 | 0 | 96.8%<br>3.2% |
| G#6 | 0 | 0 | 0 | 0 | 0 | 0 | 88 | 56 | 61.1%<br>38.9% |
| A6 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 116 | 96.7%<br>3.3% |
| | 86.8%<br>13.2% | 6.3%<br>93.7% | 25.6%<br>74.4% | 14.4%<br>85.6% | 23.7%<br>76.3% | 17.1%<br>82.9% | 10.0%<br>90.0% | 13.2%<br>86.8% | 24.64%<br>75.36% |

Figure C.6: Confusion Matrix from Bells Pitches Test using ECOC and on dFCRBM Synthesis of 4096/2048 FFT

Table C.4: Resulting accuracies of ECOC on tested models

| **MODEL** | VIOLIN SCALE | OBOE SCALE | BELLS SCALE |
|---|---|---|---|
| $D_{r1}$ | .30/.40 | .21/.17 | .18/.62 |
| $D_{c1}$ | .48/.60 | .34/.39 | .17/.16 |
| $D_{a1}$ | .16/.12 | .28/.21 | .13/.15 |
| | | | |
| $D_{r5}$ | .27/.33 | .26/.16 | .29/.67 |
| $D_{c5}$ | .44/.52 | .50/.43 | .16/.17 |
| $D_{a5}$ | .38/.52 | .19/.14 | .14/.18 |
| | | | |
| $D_{r20}$ | .22/.36 | .27/.16 | .30/.63 |
| $D_{c20}$ | .44/.54 | .38/.35 | .16/.17 |
| $D_{a20}$ | .36/.57 | .38/.34 | .14/.17 |

Table C.5: Resulting accuracies of ECOC/NB on $D_{model}$

| **MODEL** | VIOLIN SCALE | OBOE SCALE | BELLS SCALE |
|---|---|---|---|
| $D_{r1}$ | .30/.40 | .21/.17 | .18/.62 |
| $D_{c1}$ | .48/.60 | .34/.39 | .17/.16 |
| $D_{a1}$ | .16/.12 | .28/.21 | .13/.15 |
| $D_{r5}$ | .27/.33 | .26/.16 | .29/.67 |
| $D_{c5}$ | .44/.52 | .50/.43 | .16/.17 |
| $D_{a5}$ | .38/.52 | .19/.14 | .14/.18 |
| $D_{r20}$ | .22/.36 | .27/.16 | .30/.63 |
| $D_{c20}$ | .44/.54 | .38/.35 | .16/.17 |
| $D_{a20}$ | .36/.57 | .38/.34 | .14/.17 |

# REFERENCES

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[2] S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data-glove and a speech synthesizer," *IEEE transactions on Neural Networks*, vol. 4, no. 1, pp. 2–8, 1993.

[3] S. Fels and G. Hinton, "Glove-talkii: an adaptive gesture-to-formant interface," in *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM Press/Addison-Wesley Publishing Co., 1995, pp. 456–463.

[4] A. Hunt and R. Kirk, "Mapping strategies for musical performance," *Trends in Gestural Control of Music*, vol. 21, pp. 231–258, 2000.

[5] J. Pressing, "Cybernetic issues in interactive performance systems," *Computer music journal*, vol. 14, no. 1, pp. 12–25, 1990.

[6] M. M. Wanderley and N. Orio, "Evaluation of input devices for musical expression: Borrowing tools from hci," *Evaluation*, vol. 26, no. 3, 2006.

[7] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer music journal*, vol. 26, no. 3, pp. 11–22, 2002.

[8] N. Schnell, *Playing (with) Sound-Of the Animation of Digitized Sounds and their Reenactment by Playful Scenarios in the Design of Interactive Audio Applications.* na, 2013.

[9] R. M. Neal, "Connectionist learning of belief networks," *Artificial intelligence*, vol. 56, no. 1, pp. 71–113, 1992.

[10] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[11] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[12] G. Mayraz and G. E. Hinton, "Recognizing handwritten digits using hierarchical products of experts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 189–197, 2002.

[13] V. WhyeTeh and G. E. Hinton, "Rate-coded restricted boltzmann machines for face recognition," 2001.

[14] G. W. Taylor, "Composable, distributed-state models for high-dimensional time series," Ph.D. dissertation, University of Toronto, 2009.

[15] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, p. 0, 2006.

[16] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in boltzmann machines," *Parallel Distrilmted Processing*, vol. 1, 1986.

[17] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," DTIC Document, Tech. Rep., 1986.

[18] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning.* ACM, 2007, pp. 791–798.

[19] R. Memisevic and G. Hinton, "Unsupervised learning of image transformations," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 2007, pp. 1–8.

[20] R. Memisevic, "Non-linear latent factor models for revealing structure in high-dimensional data," Ph.D. dissertation, University of Toronto, 2008.

[21] J. Burnham, "Systems esthetics," *Artforum*, vol. 7, no. 1, pp. 30–35, 1968.

[22] J. Burnham, "The aesthetics of intelligent systems," *On the future of art*, p. 119, 1970.

[23] J. Burnham, "Software: Information technology: Its new meaning for art," *Jewish Museum, New York*, 1970.

[24] A. Razdow and P. Conly, "Composer," *Jewish Museum, New York*, 1970.

[25] D. Antlin, "The conversationalist," *Jewish Museum, New York*, 1970.

[26] S.-K. I. of Visual Sciences, "Vision substitution system," *Jewish Museum, New York*, 1959.

[27] W. Vandouris, "Light pattern box (electrochrome)," *Jewish Museum, New York*, 1970.

[28] H. Haacke, "News," *Jewish Museum, New York*, 1970.

[29] H. Haacke, "Visitor's profile," *Jewish Museum, New York*, 1970.

[30] L. Skrebowski, "All systems go: Recovering hans haacke's systems art," *Grey Room*, no. 30, pp. 54–83, 2008.

[31] L. V. A. Center, "Hans haacke 1967," 2011.

[32] L. Jevbratt, "Interspecies collaboration: Making art together with nonhuman animals." [Online]. Available: http://jevbratt.com/writing/jevbratt_interspecies_collaboration.pdf

[33] L. Thomas, F. Amini, and R. Lannon, "A general theory of love," 2000.

[34] I. Xenakis, *Formalized music.* Indiana University Press, 1971.

[35] K. Saariaho, "Lohn," 1996. [Online]. Available: http://saariaho.org/works/lonh/

[36] R. Nieminen, "Io, program notes," 1987. [Online]. Available: http://saariaho.org/works/io/

[37] R. Nieminen, "Six japanese gardens, program notes," 1994. [Online]. Available: http://saariaho.org/works/sixjapanesegardens/

[38] K. Saariaho, R. Malka, J.-B. Barrière, J.-B. Mathieu, I. Stoïanova, D. Upshaw, A. Karttunen, C. Hoitenga, and F. Jodelet, *Prisma.* Montaigne Naïve, 2001.

[39] S. Gresham-Lancaster, "The aesthetics and history of the hub: The effects of changing technology on network computer music," *Leonardo Music Journal*, pp. 39–44, 1998.

[40] J. Bischoff, "The hub," 1989.

[41] J. Bischoff, "The glass hand," 1996.

[42] R. Kuivila, "Open sources: Words, circuits and the notation-realization relation in the music of david tudor," *Leonardo Music Journal*, vol. 14, pp. 17–23, 2004.

[43] D. Tudor and D. Tudor, *David Tudor: Neural Synthesis.* Lovely Music, Limited, 1995.

[44] F. Warthman, "David tudor: Nerural synthesis, program notes," 1995.

[45] P. Duhamel and M. Vetterli, "Fast fourier transforms: a tutorial review and a state of the art," *Signal processing*, vol. 19, no. 4, pp. 259–299, 1990.

[46] A.-r. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on.* IEEE, 2011, pp. 5060–5063.

[47] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[48] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[49] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[50] M. ali Bagheri, G. A. Montazer, and S. Escalera, "Error correcting output codes for multiclass classification: application to two image vision problems," in *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on.* IEEE, 2012, pp. 508–513.

[51] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 289–302, 2014.

[52] G. Zheng, Z. Qian, Q. Yang, C. Wei, L. Xie, Y. Zhu, and Y. Li, "The combination approach of svm and ecoc for powerful identification and classification of transcription factor," *BMC bioinformatics*, vol. 9, no. 1, p. 282, 2008.

[53] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning.* Springer series in statistics New York, 2001, vol. 1.

[54] D. M. Christopher, R. Prabhakar, and S. Hinrich, "Introduction to information retrieval," *An Introduction To Information Retrieval*, vol. 151, p. 177, 2008.

[55] A. Horner and D. E. Goldberg, "Genetic algorithms and computer-assisted music composition," *Urbana*, vol. 51, no. 61801, pp. 437–441, 1991.

[56] J. B. Mailman, "Agency, determinism, focal time frames, and processive minimalist music," pp. 125–43, 2013.

[57] S. Cuykendall, "a performer's perspective." [Online]. Available: http://performersperspective.movingstories.ca/about/

[58] C. Roads, *Microsound*. MIT press, 2004.

[59] M. J. Junokas and S. Cuykendall, “a performer’s perspective: translation of three improvisers.” [Online]. Available: http://performersperspective.movingstories.ca/translational/

[60] S. Fine, Y. Singer, and N. Tishby, “The hierarchical hidden markov model: Analysis and applications,” *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998.

[61] E. Ulman, “Some thoughts on the new complexity,” *Perspectives of new music*, pp. 202–206, 1994.

[62] R. Toop, “On complexity,” *Perspectives of new music*, pp. 42–57, 1993.

[63] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[64] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle et al., “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[65] G. W. Taylor, G. E. Hinton, and S. T. Roweis, “Modeling human motion using binary latent variables,” *Advances in neural information processing systems*, vol. 19, p. 1345, 2007.

[66] Y. Bengio et al., “Learning deep architectures for ai,” *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[67] C. Nvidia, “Compute unified device architecture programming guide,” 2007.

[68] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with cuda,” in *ACM SIGGRAPH 2008 classes*. ACM, 2008, p. 16.

[69] J. Reese and S. Zaranek, “Gpu programming in matlab,” *MathWorks News&Notes. Natick, MA: The MathWorks Inc*, pp. 22–5, 2012.

[70] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[71] Y. Nesterov, “A method of solving a convex programming problem with convergence rate o (1/k2),” in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[72] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.