A machine learning approach to computer modeling of musical expression for performance learning and practice

Fábio José Muneratti Ortega

TESI DOCTORAL UPF / 2021

Director de la tesi

Prof. Dr. Rafael Ramírez Melendez,
Department of Information and Communication Technologies

To mom and dad.

Acknowledgements

My best achievement in this work is to have so many people to thank, starting from Rafael, my supervisor, for having the trust and providing the means to bring me to this point. I also want to extend my gratitude to all UPF staff who successfully defused my frequent states of bureaucratic armageddon: Sonia, Cristina, Montse, Jana, Vanesa, Federico, and especially Lydia, who reassured me that I would find at least one friendly person in Barcelona.

To my partners in teaching, especially Patricia Santos, Montse Fernandez and Toni Lunar, I want to thank for having your amazing company and for being exceptional professional role models for me.

My dear thank you to the lovely cast of linguists (with a few additions from other fields) who first managed to lure me away from PhD work, and much further still from loneliness: Mayyar, Gang(-Gang), Sylvia, Zi, Vito, Leticia. Domi, not being able to run into you and go for coffee was the worst part of working remotely. Rebecca, thanks for bringing joy back to our home earlier this year (and for all the wine, too). Jacopo, I'm hoping we can find ourselves some more philosopher meet-and-greet events soon. Meztli, my salsa moves are not the same without your help.

Thank you also to all friends from the MTG, the official members and the honorary ones: Albin, Pritish, Xavier, Cecilia, Tessy, Oussam, Roisin, Jordi, Minz, Olga, Marius, Sertan, Pablo Zinemanas, Rong, and Xavier Serra for welcoming me in this talented group. I have grown as a person and as a scientist from the conversations with each one of you. A special thank you goes to my jam partner Seva, who really put the music in

music technology, and formed a bond with me far beyond the tunes. Likewise, thank you to Alia, Colm, Jyoti, Nazif, and everyone who eventually joined us for the pleasure of our ears and hearts.

In particular, I am most grateful to the friends in the Music and Machine Learning Lab. Thank you Magdalena, Zach, Batu, Óscar, Pablo, Gil, and Tetsuro. Alfonso, your input and assistance dug me out of multiple holes. David, you were a multidisciplinary teacher to me. Stay sharp on the chess board, or next time I (or Jordi) will get you. Àngel, your friendship, daily company and wisdom were the peak of PhD routine, and there is no doubt that this thesis would be very different – for worse – without you. Last but not least, I thank Sergio for probably all of the above! You are an incredible example of teacher, researcher, musician and friend. Send hugs to Yeliz, Camilo and Mateo for me.

I can't leave out a big hug to Dani and Isra. You guys were our honorary neighbors and catalan family all at once, and we already miss you.

To the friends from back home, Earth wasn't big enough to prevent your love from reaching me. Fabio and Isa, your partnership means the world. Ligia, so much of myself I owe to you. Bel, you helped more than you know. Paula, Talita, PH, Gu, Wil, Vitória, Lara, Runas, Pedro and everyone from Santana, thanks, and I'll be seeing you soon.

I am also deeply thankful to everyone who participated in my experiments, in the absurd situations I put you through for nothing else than friendship and a taste for science and music.

And, of course, putting the words on paper is a really small fraction of what it took to complete this work. The real accomplishment belongs to my family. With every hardship, big or small, the only reason I move forward is their support, and the joy in my victories I see in their eyes. Vicente, my father, Vânia, my aunt, my grandmothers Diva and Idalina – this work is more theirs than it is mine.

Mom, no one was looking forward to my finishing as much as you. These lines reflect who you were and all you taught me as best as I could. I will share your guidance and your love in your honor, always.

Marina, I really put you through a lot these years. It was under the shadow of this work that we learned to share our lives, and that reason alone made it worth it. Thank you for your wisdom, for your love, for keeping me human, and pushing me to be a better one.

Abstract

This thesis deals with the design and implementation of computer systems for expressive music performance (CSEMP), exploring different methods from machine learning and reflecting on the role of musical structure in the emergence of performance patterns, as well as the applicability of each approach in a pedagogical setting. Three models are described and evaluated: a lazy learning approach using a phrase similarity measure, an evolution of the previous with parameterized performance features, and a deep-learning model with sequential encoding of musical information. Results demonstrate that the simpler phrase-level approaches can generate stimulating performances with small datasets, and that the deep-learning approach can achieve high accuracy predicting performance information. Their analyses also highlight the challenges of designing systems for instruments beyond the piano. The pedagogical potential of technologically-enhanced settings is addressed with the proposal and pilot evaluation of a performance practice method using the SkyNote software.

Resum

Aquesta tesi tracta sobre el disseny i la implementació de sistemes informàtics per a l'execució musical expressiva (CSEMP), explorant diferents mètodes de l'aprenentatge automàtic i reflexionant sobre el paper de l'estructura musical en el descobriment de patrons d'actuació, així com l'aplicabilitat de cada sistema en un entorn pedagògic. Es descriuen i s'avaluen tres models: el primer d'ells utilitza una mesura de similitud de frases; el segon, una evolució de l'anterior amb característiques d'actuació parametritzades; i l'últim, un model d'aprenentatge profund amb codificació seqüencial de la informació musical. Els resultats demostren que els enfocaments més senzills a nivell de frase poden generar actuacions estimulants amb conjunts de dades petits i que l'enfocament d'aprenentatge profund pot aconseguir prediccions d'alta precisió sobre la interpretació de peces musicals. Les seves anàlisis també destaquen els reptes de dissenyar sistemes per a instruments més enllà del piano. El potencial pedagògic dels entorns tecnològicament millorats s'aborda amb la proposta i l'avaluació pilot d'un mètode de pràctica d'actuació mitjançant el programari SkyNote.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> As far as the execution is concerned… the most frequent and most serious mistake is to follow the music instead of preceding it.
>
> Nadia Boulanger

1.1 Motivation

Modern music notation is a powerful tool, with immense impact in the preservation and communication of musical ideas and compositions. Music performance, however, is far more complex than what the notation represents, leaving gaps in our ability to communicate its elements and, as a consequence, understand them and teach them.

Expression is a key aspect of what makes music performance pleasurable. By playing in particular manners, musicians can hold the attention of listeners, elicit emotions, facilitate comprehension of a piece, and communicate artistic intentions.

In the context of this work, the term *expression* in music refers to the elements of a musical performance that depend on personal response and that vary between different interpretations (Baker et al., 2001). These elements can be identified by the sonic features they produce, and are, in many cases, characterized by how they cause the music to deviate from a neutral, or as we shall refer to, "deadpan" performance of a musical

composition (Palmer, 1997; Seashore, 1938). For every musical instrument, several such elements, or *expressive performance actions* (EPAs), may be shaped by the performer, such as tempo, dynamics, articulation, and so on.

Music students and teachers alike have pointed to expression as the most valued skill in performers (Laukka, 2004; Lindström et al., 2003a), however, teaching this skill presents several challenges, and often ends up neglected in classrooms (Karlsson and Juslin, 2008; Meissner, 2017).

In an influential study about expression teaching practices, Woody (2006) compared three instructional approaches used to elicit expressivity in music students' performances: aural modeling, instruction about concrete musical properties, and instruction using metaphors and imagery. His results were able to validate the effectiveness of all three while highlighting advantages and disadvantages of each.

While metaphors and imagery prompted the greatest variation in students' performances, those variations were not necessarily leading them closer to the expert's performance which originated the metaphorical feedback. This is consistent with the observation by Juslin et al. (2006) that such imagery is inherently ambiguous since it relies on performers' personal experiences. On the other hand, that same ambiguity may give performers more leeway to come up with their own interpretations, as many music teachers encourage (Meissner, 2017).

The aural modeling method led to performances more consistently similar to the model, though large changes in performances weren't always observed. The concrete instructions approach succeeded in inducing more practice, though not reliably in improving overall performance. Interestingly, in a survey among musicians about their preferred approaches, Bonastre and Timmers (2021) report that concrete, technical directions ranked the highest, and modeling, the lowest among the discussed techniques for teaching adults. The authors discuss that one limitation of aural modeling is that the student must be capable of extracting the useful information from the performance, that is, they must know what to listen for in the model performance.

Given these pedagogical challenges, technology may offer new approaches combining the positive aspects and overcoming some limitations of the currently used methods for expressive performance practice. It is now viable and relevant to investigate scenarios in

which students would practice performance equipped with a computer system that is able to provide visual feedback on their expressive performance actions as well as contrasting them to another, referential performance. This type of setting has potential to enhance the teaching methods discussed above as it combines modeling with concrete information that breaks down the elements of the model performance into concrete directions, attacking the issue raised by Bonastre and Timmers (2021). Additional support to this scenario can be seen in the results obtained by Lisboa et al. (2002), that revealed long-term improvement in musicians' expressivity after studying by means of performance imitation, and also in the defense of real-time visual feedback (RTVF) for expressive music learning found in Sadakata et al. (2008).

Even though this new approach presents many advantages, the potential of computer-assisted music learning settings extends beyond their application as improved aural modeling tools. Our primary interest in the investigations reported here is to explore the use of computers for their data analysis capabilities, assisting musicians in realizing patterns of performance and shaping their expressive abilities according to more specific and creative goals.

This work discusses the development of computer systems for expressive music performance (CSEMP) – that is, computer programs designed to generate EPA information given a certain musical context – and their application as pedagogical tools for expressive performance learning.

Whereas a conventional imitation exercise presents an individual musicians' interpretation as a blueprint, if a computer generated performance is used instead, that model can be built around data from several different musicians, thus offering the student access to a representation of the collective expectation around a piece. The mere contrast between a musicians' EPAs and the computer system's inferences about it can be a source of inspiration for challenging one's assumptions about elements of performance that might otherwise be taken for granted.

Another improvement gained with CSEMP for expressive learning is that the computer analyses, its model suggestions and visualizations can assume a role of mediation between student and teacher, encouraging discussion about the character of a performance and the translation of intentions and emotions into technique. Students can

also be prompted to offer their critique of the generated performances, thus fostering critical thinking about interpretations in a low pressure environment – as no particular performer would be criticized. Dialogic approaches have been shown to be very effective for developing students' expressive skills (Meissner et al., 2020; Meissner and Timmers, 2019) and the inclusion of computer tools opens an opportunity for doing so in a systematic way.

Lastly, technology-enhanced learning of music expression has potential to improve self-study. Typical methods of teaching expression rely heavily on continuous feedback from master to apprentice. As a consequence, students are left with limited tools for improving in that field when practicing on their own, a setting which makes up the bulk of any musician's practice time. If a computer model can be trained to provide relevant expressive suggestions in a given musical context, it could further stimulate students' perceptions and improvement even in the absence of an expert.

1.2 Goals

The main objective of the research presented herein is to propose and evaluate computer systems that model musicians' expressive actions in performance so that the resulting models are applicable as tools for learning and practicing music expression.

This goal is further specified and limited in its scope by some additional premises.

The modeling strategies that were explored belong to the machine-learning field, therefore the problem of generating realistic EPAs is viewed as problem of optimization and data from real performances are incorporated in the design as observations of a target function. This is a deliberate design decision since it provides a direct relationship between empirical observations of performance patterns and the resulting models, minimizing the designer's personal artistic influence in the model outputs.

The systems were developed and analyzed targeting the performance of classical music of the western canon, as this is the musical tradition that offered us the largest amount of available data in recordings, musical theory, and local practitioners.

Most significant to the direction of the research is that the modeling strategies were intended to be applicable to a wide range of musical instruments. This is in sharp contrast

to the majority of the existing body of knowledge in the field (see chapter 2) which focuses primarily on the piano. As a compromise, we have decided to focus on the generation of signals from two classes of expressive performance actions: *timing* – which consists of the variations in note onset time, note durations, and tempo along a piece –, and *dynamics* – as represented by proxy features such as loudness or velocity of movement of sound-producing instrument parts. These two elements are present in the performance of most musical instruments, and can be inferred from audio recordings alone. The instrument chosen for evaluation in most scenarios presented ahead has been the violin.

Our research goals as stated bear a few direct consequences worth indicating. Notably, the application of the developed models of performance to a computer-assisted study of expression implies that not all information about expressive actions are equally important. Information related to deliberate, long-term actions take precedence over unconscious, short-term variations in sound qualities, since it is mostly learning the former that interests a studying musician, whereas the latter is a natural by-product of a human performance. We can build an analogy between the desired outcome of our system and the information conveyed to musicians by an orchestra conductor – the type of expressive direction which is reasonable to communicate during a performance is related to an overall character rather than specific sound features, and it is slowly evolving through time, rather than being specific to every note. Nevertheless, distinguishing the different sources and causes for expressive variations in music is itself a challenge to be tackled.

Finally a secondary but important objective, given our interest in the application of CSEMP for the learning and practice of expression, is to evaluate the impact of a technology-enhanced setting in a performance learning task.

Having presented the main context, we can summarize the scope of our work in terms of the following research questions:

- Is it possible to design models that generate EPA signals which are useful for the practice of music expression, particularly for instruments other than the piano?
- With current state-of-the-art algorithms, what are the most relevant features and design decisions for modeling expressive performance?

- Is it possible to facilitate the practice of music expression with the help of a technologically enhanced setting?

1.3 Contributions

The wide scope encompassed by this research allowed us to achieve several modest contributions.

- A *phrase modeling approach* to expressive music performance generation was developed and evaluated, with implications to the mathematical treatment of dynamics and psychological perception of violin sounds as highlights.
- An *expressive solo violin performance dataset* was created from scratch, including the recording of 81 musical excerpts, machine-readable score transcriptions, and manual audio-to-score alignment.
- Another dataset, consisting of *Beethoven violin sonatas*, was derived from the MusicNet (Thickstun et al., 2017) recordings, with the addition of measure boundaries information for all available movements.
- An approach to *time-series generation via deep-learning sequence models* was developed, with innovations in the treatment of input sequence segmentation and encoding for representation learning.
- As a consequence of the previous, state-of-the-art accuracy in *expressive feature prediction* from scores without expressive notation was achieved, demonstrating the importance of phrase structure to musical interpretation.
- A systematic study on the *impact of deep-learning design elements* to the problem of music performance generation is provided, which should hopefully assist future research decisions.
- Finally, our secondary goal was achieved via the design and implementation of a technology-assisted music performance method of practice in the form of new functionality included in the SkyNote software (Ramirez et al., 2018).

- Complementing the previous, a pilot study on the reception and impact of such methodology to the learning of expressive performance on the violin is also reported.

1.4 Thesis Outline

The following chapter discusses previously developed CSEMP, as well as some recent advances in deep learning applicable to our problem and finishes with a review of the studied approaches for technology-enhanced music learning. Chapter 3 presents our results with phrase-level modeling of expression based on melodic similarity. Chapter 4 introduces the generalized version of that model and the results of its evaluation. Chapter 5 discusses our note sequence model, and presents its evaluation and contrast to the previous approach. Chapter 6 then shifts attention to our performance practice method proposal and implementation within the SkyNote software, along with its pilot evaluation. Finally, chapter 7 presents final remarks and the thesis conclusions.

Chapter 2

Background

2.1 Computer Systems for Expressive Music Performance

Definitions

Expression in musical performances has been an active field of study for some time. Researchers have approached the problem from several perspectives, from the measurement and classification of performance patterns to the search of parallels between structural musical features in a composition and performance actions (Widmer and Goebl, 2004; Palmer, 1997; Gabrielsson, 2003). Their motivations can be placed in two broad categories: analytical and synthetical (Cancino-Chacón et al., 2018); when viewed from an analytical standpoint, computer models are used as tools for gaining insight into the way humans perform music – this is the case for the pedagogical application discussed earlier. Nevertheless, a significant amount of research in the field targets performance synthesis, as that also finds a range of applications such as producing realistic renditions within score transcription software, improving MIDI file playback, and even providing expressive automatic accompaniment for musicians, among others.

Cancino-Chacón et al. (2018) define computational models of expressive performance as "attempts at codifying hypotheses about expressive performance – as mappings from score to actual performance – in such a precise way that they can be implemented as computer programs and evaluated in systematic and quantitative ways". Glaring in that definition is the fact that nothing is said about the nature of such hypotheses, and indeed

it is possible to distinguish them from a number of criteria (e.g.: under what conditions do the hypotheses apply, whether their predicted effects are relative or absolute, whether they are probabilistic or deterministic etc.). Chief among them for our purposes is the distinction between *rule-based* and *data-driven* models. The hypotheses present in rule-based models establish direct links between composition and performance whereas in data-driven models they are mediated by empirical evidence. As a simplistic example for the sake of clarification, if a rule-based hypothesis were formulated as: *"Performed loudness increases with pitch at a rate of 2dB per octave for pitches higher than G4."*, a similar counterpart in a data-driven model would be: *"Loudness and pitch are linearly related at the rate that best approximates the examples in the given dataset."*.

In that sense, it can be said that an overarching hypothesis common to all data-driven models is that the collection of recurring patterns present in the dataset of performance examples constitute the ideal reference, their differences being how they sample the universe of musical performances and what mathematical structures are allowed in crafting the score-to-performance mappings.

To delve deeper into the details of CSEMP, we borrow from the generic model devised by Kirke and Miranda (2009) and reproduced here in figure 2.1.

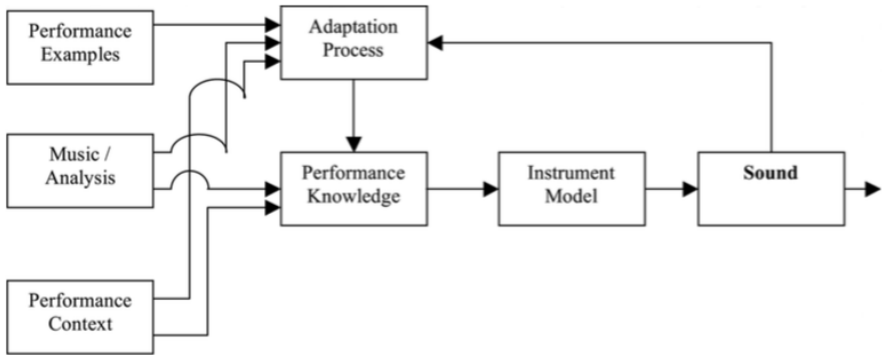


Figure 2.1: Kirke and Miranda's model of CSEMP

The diagram, although designed to represent commonalities among those systems, reflects specifically an application in performance synthesis whereas we are primarily interested in an analytic application involving performance suggestions and feedback. To ac-

count for that difference in scenario, the instrument model and sound modules should instead be replaced by a more generic *performance knowledge parser*, that would, in our case, decode the performance knowledge into tangible, humanly-comprehensible information about concrete expressive performance actions. This, however, doesn't preclude the application of such knowledge in a synthesis engine much in the same vein as the two replaced modules suggest.

The central and most important element in a CSEMP is represented in the Performance Knowledge module, which is where the system's "expertise" is encoded, be it via learned parameters in a machine learning algorithm or by rules of performance programmed directly by the designers.

The input to the system comes through the Music/Analysis module, and that is where significant design decisions regarding the representation of musical information are stored, including, if it is the case, an analytical breakdown of musical structure. Many of our contributions to modeling are found in this module, as we dedicated much effort to exploring the impact of the input encoding format to the quality of the generated output.

The Performance Context block represents additional information supplied to the system beyond the music itself, such as the mood of the performance, e.g.: lively, melancholic, aggressive, or if it adheres to a certain style, e.g.: baroque or romantic. This addition enables generating multiple variations in performance for the same musical piece in a way that preserves causal relationship to the inputs, giving more creative control to the users of the system.

The final two modules, Performance Examples and Adaptation Process, represent the usage of past human performances to inform and perfect the performance knowledge of the system. This is a typical setup in supervised machine learning systems, but the authors emphasize that the work represented in the adaptation module can also be partially or entirely manual, as the designers monitor the generated performances and fine tune the Performance Knowledge module to produce results that better match their past experiences and expectations.

We have established a distinction between rule-based and data-driven models of expressive performance in terms of how the premises for performance generation originate.

Some of the most influential early models of music performance were of the former kind, and the best example is the KTH model.

The KTH model (Sundberg et al., 1983), implemented in the *Director Musices* software (Bresin et al., 2002) is a rule-based performance model continuously developed at the Royal Institute of Technology in Stockholm over many years. With it, performance actions can be predicted based on directly stated rules relating them to score features. We can attribute its success and longevity to the clear scope of each rule and the possibility of controlling the intensity with which each one of them is applied. Though the rules have been derived from empirical research, it is not a learning or data–driven system, but rather a collection of theoretical principles which don't claim to be necessary or sufficient to explain the expressivity of a given performance.

Also relevant as seminal theoretical models is the work of Todd (1985); McAngus Todd (1992) which provide evidence that timing and dynamics – the primary dimensions of expressivity in piano performances, as first pointed out by Seashore (1938) – tend to be coupled to the structure of musical pieces, and also the work of Clynes (1995) that observes how the interpretation of pieces from specific composers may converge to particular styles of performance that are preferred by listeners for their pieces but not for those of other composers.

Data-driven approaches

As machine learning algorithms rose as a viable alternative to modeling a wide range of tasks, many examples of learning CSEMP were developed. This type of CSEMP is particularly interesting to our scenario since the resulting outputs are empirically backed by virtue of the very formulation of the system, and can reflect different musical styles simply by training it with different sets of examples. Although, to the best of our knowledge, no expression models have been proposed with a pedagogical application in mind and only a minority deal with the violin, their design and methodology are applicable to this research. For a thorough and comprehensive review of CSEMP literature, we refer the reader to Cancino-Chacón et al. (2018). Next is a short overview of the most significant research for giving context to our own contributions and the outlook of the field.

A general trend that can be highlighted in the progress of CSEMP research, as in other machine-learning dominated fields, is a growing emphasis on acquiring and processing larger datasets. As a consequence of that, whereas in earlier systems much work was put into parsing the musical information present in the performance examples, as researchers moved towards systems prepared to learn from a wider range of pieces, this responsibility shifted to the learning algorithm, so that a bigger share of the performance knowledge observed in such systems is derived directly from the performance examples, and less so from the designers themselves and how they choose to structure the generation process.

The systems proposed by Tobudic and Widmer (2006, 2003); Widmer and Tobudic (2003) are good examples of CSEMP designed to leverage performance knowledge obtained from a modest number of samples. Their final version, named DISTALL, was evaluated with a training set consisting of only 15 Mozart piano sonata movements with positive results, particularly for generation of dynamics. For their approach, pieces were hierarchically divided into three levels of phrases by a musicologist, and a measure of phrase similarity was defined based on melodic and harmonic features of the phrases, organized via a first-order logic language. This similarity measure is then used during generation to select the best match in the training set for every input phrase with a nearest-neighbor algorithm. The outputs produced are a combination of the contributions from each hierarchical level, with information from the two higher levels parameterized as a second-degree polynomial and the lowest, note-level information – which the authors refer to as "residual" – being generated using rules from an earlier system, the PLCG (Widmer, 2003).

The phrase-level and instance-based approach chosen by the authors is conceptually very appropriate to smaller datasets, because by copying EPAs from the selected instances verbatim, it takes full advantage of the idiomatic vocabulary of expressive actions available in the sampled sources, even though the number of available samples might be insufficient for a CSEMP to develop such a vocabulary from scratch. Their dependency on manual segmentation of musical pieces, however, stands out as a limitation, since it injects significant structural interpretation of scores, and prevents fully automated expressive performance generation from symbolic music alone.

YQX (Flossmann et al., 2013) is a system from the generation of CSEMP that followed

DISTALL, and it further illustrates the research trend that we observed. Many combinations of pieces were used for its training and evaluation, coming from two corpi of performances by renowned pianists using Bösendorfer SE pianos, which are capable of registering precise keystroke timing and velocity information. Together, the Mozart corpus and the Magaloff corpus (Flossmann et al., 2010) sum over 15 hours of playing time.

In terms of its design, YQX employs Bayesian network theory to model interactions between score and performance. Score information is represented by feature vectors of both discrete and continuous variables on the note level, quantifying melody and rhythm, whereas performance information corresponds to the note-level signals from the Bösendorfer SE, numerically conditioned to represent timing, velocity, and articulation EPAs. Given its Bayesian nature, the system achieves performance generation by computing estimates of the probabilities of the outputs conditioned by the inputs or, in other words, generating the most likely performance given its observations, under general assumptions of independence among variables and gaussian behavior. This logical and straight-forward generation strategy makes the system easier to reproduce and interpret. Because performance generation is done on a note-level only, the system naturally lacks context-awareness. To compensate for that, some input features relate notes to their context by means of musicological concepts such as consonance and dissonance in tonal theory and Eugene Narmour's Implication-Realization model (Gjerdingen and Narmour, 2006). Even more significantly, in its most sophisticated instance, YQX generates performances by maximizing joint probabilities for note sequences, effectively giving the model awareness of past outputs.

On the one hand, YQX has great scalability with respect to training set size. The required calculations are feasible with larger datasets, and the system's operation is entirely automated. On the other hand, its design is limited in what it can represent. The probabilistic rationale doesn't easily incorporate complex performance patterns or conflicting performance styles in training.

Another, more recent, example of probabilistic CSEMP is by Moulieras and Pachet (2016). Like the previous examples, it is also a system for piano performance modeling, though the musical corpus used for training and evaluation consists of jazz and popular pieces instead of western classical ones, with a total of 172 songs played by the

same pianist. The approach consists of estimating probability distributions for the EPA random variables following the principle of maximum entropy. The parsing of musical information is minimal, as the metrical position of notes in the measure is the only score information reportedly collected. A theoretical benefit of the maximum entropy modeling design is that all performance features are jointly modelled, favoring greater coherence among the variables then when generated independently. The system operates on the note-level, with limited contextual information arising from constraints related to a window of three notes before and after each target note, mildly enhanced by the iterative nature of the generation process.

If we consider the types and formats of input and output information, the various models of expressive performance found in the literature also differ significantly in scope, making a comparative evaluation or benchmark creation very hard. The Basis Functions model by Grachten and Widmer (2012) serves to illustrate this point. Their system attempts to provide a solution to a much more specific question about music performance: given a musical score containing some performance guidance in the form of expressive markings, in what way do musicians interpret such markings when playing? The problem is very practical, since the inclusion of expressive markings in scores is common practice, and even though the meaning of each symbol is well defined, their translation into sound is very subjective and ambiguous. The approach consists of interpreting each type of marking as specific mathematical functions of timing or dynamics and, based on training examples, optimizing the weights of each function for combining them into prediction values. Developing this model further, Gadermaier et al. (2016) successfully applied it to predictions of dynamics in orchestral ensembles, and Cancino-Chacón et al. (2017) generalized the system for non-linear combinations of basis functions. In spite of the great results obtained by these systems, the more general problem of performance generation in the absence of such "hints" as score markings persists, and requires different modeling approaches.

Even though systems for piano performance generation have seen gradual improvements over the years, existing research on other, more challenging instruments is more akin to earlier works for piano: smaller-sized training datasets compensated by carefully crafted features with clear musical meaning.

Ramirez et al. (2008) design a model for jazz saxophone that produces performance rules

based on data via genetic algorithms. The evolutionary aspect of their method is meant to select the sets of logical rules that best explain the expressive features in sample performances, therefore producing knowledge abstractions that possess musicological meaning, and can be taught directly.

Though not designed for modeling expression, the system described by Maestre (2009) is meant to predict violin bow motion directly from score notation, taking the limited bow length and bowing techniques into consideration. The intention of the model was to drive physical models of synthesis, but since all expressive actions are simply a consequence of the musician's motion, the same principle can be applied for offering expressive performance guidance. The set of input features of the system include information about the desired type of violin articulation and dynamics (e.g.: *detaché*, bow down, *pianissimo*) so, as was the case with the Basis Functions model, it doesn't generate expressive content from scratch, but provides a mapping from high-level, possibly unclear instructions, to a well-defined performance description.

Giraldo and Ramirez (2016) tackled jazz guitar performance in their system and went beyond the generation of timing and dynamics EPAs by also including presence or absence of ornamentation as an output. Their training data consisted of 16 pieces, and the input feature set extraction was comparable to YQX in concept. Interestingly, unlike most CSEMP, which produce EPA signals by means of regression, they opt for discretization of the outputs, generating note level indications such as *piano* and *shortened* duration.

The examples presented so far give an overview of data-driven CSEMP which have made use of what we may call *conventional machine-learning* (LeCun et al., 2015). The development of its counterpart, *deep learning*, introduces a shift in design which we will explore more deeply in the following sections. Regardless of the chosen design methodology, though, it is clear that expressive performance generation is a difficult problem to generalize, and even more difficult to evaluate. Even though these systems are trained to better predict the expressive elements of performance, that goal is virtually unattainable, and thus their predictive ability is not really a good metric of quality. The most logical substitute, perceptual evaluation, is contingent on the population sample, not very scalable, difficult to reproduce, and doesn't contribute to forming a benchmark to measure future systems against. In short: a complex field, moving forward in modest steps.

Deep-Learning Models of Musical Language

In a significant review article for Nature, LeCun et al. (2015) argue that conventional machine-learning algorithms present difficulties processing raw data, instead relying on the careful design of feature extractors by the system engineers themselves to transform raw data into suitable representations for the learning tasks. Conversely, *representation-learning* methods do exactly that, automatically learning the necessary features from raw inputs. Deep learning methods, as they define, are representation-learning methods with multiple levels of representation, going from raw data to the desired output by composing modules which operate on increasingly higher abstraction levels.

Propelled by major advances achieved by deep-learning architectures in various fields such as image recognition (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012), and machine translation (Sutskever et al., 2014), research in music generation has also become increasingly dedicated to deep-learning approaches in recent years.

The primary deep-learning architecture that enabled successful music generation systems is the recurrent neural network (RNN). RNNs are designed to process information that is organized in a sequential form, processing one element in the sequence at a time while retaining information derived from the previous elements in the form of a "state vector" (LeCun et al., 2015). This allows us to create a machine-suited representation of the music – replacing the manually crafted feature vectors in conventional machine-learning models – which is then used to generate the desired outputs. Most common tasks can thus be modeled in the form of encoders and decoders: the first layers of RNN encode the inputs as described, and the later ones learn to go from the internal representation back to an application-ready one, decoding it. In music generation systems, the output encoding matches the input, as they are typically trained to *continue* a composition: the inputs are the previous symbols representing the music, and the outputs are the symbols best fit to complete it.

The basic idea for an RNN was first developed in the 1980s (Rumelhart et al., 1986), but its application was limited due to difficulties in training and retaining meaningful information after several steps in the sequence (Bengio et al., 1994). Improvements around this issue were gradually achieved over time, most notably with the introduction of long short-term memory networks (LSTM) and later, gated recurrent units (GRU), varia-

tions of RNN developed to be able to retain short-term information learned from the input even when processing long sequences (Hochreiter and Schmidhuber, 1997; Cho et al., 2014).

A notable early example of recurrent neural networks applied to music generation is a system for generating blues improvisation by Eck and Schmidhuber (2002). The system uses long short-term memory networks (LSTM) and sequentially processes a symbolic representation of chords and melodies, each symbol representing notes on and off in a quantized time-step. The training consists of learning to estimate the probability of each note being on in the subsequent time step. Once trained, it is capable of continuing a composition, preserving the blues chord sequence and generating new melodies indefinitely.

Douglas Eck would go on to initiate the Magenta project, within Google Research, to further explore machine learning applications with music. Their Performance RNN (Simon and Oore, 2017) uses the same principle as the blues improvisation system, but with a much more powerful musical representation. The researchers encode musical information as a language with symbols to represent the start of a note, the release of a note, a shift in time, and a change in note velocity, summing up to 388 [1] different event symbols. The system is trained to compose by learning to predict the next event in the sequences of symbols representing training set pieces. By adopting this complex vocabulary, PerformanceRNN is able to parse and generate symbolic music with expressive timing – up to a precision of 10ms – and dynamics – with 32 different levels of velocity, an ability the authors analyze further in a subsequent publication (Oore et al., 2018). In order to properly learn such a rich vocabulary, many upgrades are necessary with respect to the original blues improvisation system. Most notable is the difference in the network size. The seminal model consisted of 2 layers of 8 LSTM cells each, whereas PerformanceRNN boasts 3 LSTM layers of 512 cells each. A corresponding leap in scale is observed in the training data. Whereas the blues improvisation model was trained with short melodies encoded manually by the authors, PerformanceRNN used around 1,400 piano performances from the Yamaha e-Piano Competition, which would later be organized as the MAESTRO dataset (Hawthorne et al., 2019).

[1] The original formulation quantized time in 10ms steps resulting in 388 different events. For the analysis of expressivity (Oore et al., 2018), the authors increased the granularity to 8ms steps, resulting in 413 different events.

Using much larger networks was not a revolution started by PerformanceRNN. Successfully training large-scale deep-learning models is a result of a combination of theoretical developments that improved the known methods for nearly every aspect in the systems, such as network weights initialization (Glorot and Bengio, 2010), choice of activation function (Glorot et al., 2011), regularization (Srivastava et al., 2014), and stochastic gradient descent (Kingma and Ba, 2014). Moreover, the ability to parallelize computations using graphics processing units (GPUs) was essential to making it possible to increase the complexity of such models (LeCun et al., 2015).

The paradigm adopted by PerformanceRNN and its successors in the literature views music as a language, defining a syntax for the description of musical events and then training algorithms designed for processing text with music datasets encoded in such languages.

Another example to highlight is the Music Transformer (Huang et al., 2018). This model shares the same musical representation and training set as PerformanceRNN, but structures itself around the Transformer model (Vaswani et al., 2017). The Transformer is an alternative to RNNs that has shown superior ability in capturing and reproducing long-term dependencies in sequences. Its architecture is also more parallelizable than RNNs, resulting in reduced training times. The basic building block of Transformers is the attention layer, first proposed for improving language translation tasks in RNN models (Bahdanau et al., 2015). Its principle is to provide a mechanism for learning which of the time steps from the input sequence have more influence over the current output. The authors of Music Transformer provide evidence that, like its counterpart for text, the model is more successful at producing coherent long-term structures than recurrent architectures. However, this architecture requires processing data through several attention layers, both in parallel (in a so-called multi-head attention layer) and in sequence, resulting in a large number of trainable parameters even for modest configurations, and, consequently, requiring large datasets to train properly.

All three models of music generation mentioned above share a certain limitation in their ability to generate multiple variations of compositions and in the tools they provide the user for controlling the generation, since they are trained to predict a single probability distribution for the next symbol in the sequence conditioned only by the previous symbols. Other variations of these models address that limitation, proposing more

complex approaches to generation, of which we highlight two. The first, Transformer-VAE (Jiang et al., 2020), blends the Transformer architecture with Variational Autoencoders (VAE) (Kingma and Welling, 2014) ensuring that the encoding learned by the network is a multivariate Gaussian random variable, therefore gaining more control over the input sent to its decoder portion. The second is the Adversarial Transformer (Zhang, 2020), which fine-tunes the generation process by submitting the generated sequences to a *discriminator*, a separate neural network designed to differentiate between compositions from the dataset and those generated by the Transformer network. The generator is then rewarded for "fooling" the discriminator, gradually making its music more indistinguishable from the human ones. Systems that take this approach are known in general as Generative Adversarial Networks (GAN) (Creswell et al., 2018).

We have presented a brief overview of the first effects of the introduction of deep learning in the study of musical creativity, mainly automatic composition. Deep-learning models offer a series of mechanisms to structurally take into account some of the most challenging elements in musical information, like temporally dependent relationships and data variability motivated by artistic liberties. The ability of these models to process massive amounts of data makes them capable of reproducing complex patterns that could be useful in understanding and reproducing the intricate structure of human music performance.

Even though these examples of deep-learning music systems are not CSEMP in a strict sense, as they do not provide a mapping from musical score to performance actions, their ability to process and generate musical information on a symbolic level, in some cases even including expressive content, is an indication of the applicability of their approach to our target scope. Next, we direct our attention to the existing work in that vein. For a comprehensive survey of deep music generation, we refer the reader to Ji et al. (2020).

Deep-Learning approaches to CSEMP

In contrast with music generation systems, there have been fewer proposals of CSEMP that make use of deep-learning techniques. This could be partly justified by the scarcity of large-scale datasets combining performance and symbolic music representation. Nevertheless, there are relevant examples of successful attempts, with some variation in scope.

As briefly touched upon earlier, (Cancino-Chacón et al., 2017) expanded the basis functions model to evaluate linear and non-linear models of dynamics in both piano and symphonic performances. Their evaluations include bi-directional RNN architectures of one or two internal ("hidden") layers as well as a single-layered bi-directional LSTM, besides a simpler feed-forward neural network as a baseline non-linear model, and the original linear formulation. Bi-directional RNNs effectively make the outputs aware of musical events that are yet to be performed, as well as the past ones, which is consistent with the human experience of a musician playing a known piece. Their results indicate that in CSEMP, as is the case in other fields, a larger network size and number of learnable parameters result in higher predictive power. Interestingly, the more complex recurrent architectures only surpass the performance of the simpler feed-forward one in their larger configurations, hinting that these models require more learning units to reach their full potential. Though logical, the number of experiments presented is too low to demonstrate that effect conclusively.

The design of expression generators often suffers from a similar limitation to generators of music compositions, since, in most architectures, the training process penalizes interpretations that differ from the existing examples, fixing each model's "style". The work of Malik and Ek (2017) exemplify a design that works around this issue. Their system is structured as Siamese Neural Networks, as introduced by Bromley et al. (1993). In it, two identical networks are trained in different musical styles, one in jazz and the other in classical music, though both share certain parameters which process the input, pushing the system towards a shared representation that is specialized for each case in the subsequent layers.

Yet another innovative approach to performance generation is found in Tan et al. (2020). The authors define classes of dynamics (soft or loud) and articulations (staccato or legato) for MIDI files of piano performances from the MAESTRO dataset (Hawthorne et al., 2019), and train a Gaussian Mixture Variational Autoencoder (GM-VAE) (Jiang et al., 2017) that synthesize audio performances conditioned on these expressive variables. In a sense, the authors opt for a scope similar to Cancino-Chacón et al. (2017), where the system learns how to interpret (vague) directions about expression, albeit in this case, generating the performance audio itself rather than EPA signals.

Perhaps the model that most harmoniously unites the applicable approaches previously

mentioned is VirtuosoNET (Jeong et al., 2019a). It is a system for expressive piano performance modeling based on recurrent neural networks organized hierarchically to process scores at different temporal levels. The model generates EPA signals for velocity, timing, tempo, articulation, and pedaling. Structurally, VirtuosoNET consists of a score encoder, a performance encoder, and a performance decoder.

The score encoder consists of a set of bi-directional LSTMs which process the score symbols successively, first at the note level, then at the beat, and finally measure levels. Each level takes as input the previous level outputs, combined into the corresponding timescale with the help of multi-head attention layers, in the style of Vaswani et al. (2017). The final score encoding contains one feature vector per note, formed by the concatenation of outputs from the three levels related to the note itself, its beat, and its measure.

The performance encoder uses a conditional variational autoencoder (CVAE) design, a probabilistic model first introduced by Sohn et al. (2015) which is trained to learn a latent vector representation of the "performance style", as the authors call it. This vector is modeled as a normally distributed random variable, and each performance is considered an instantiation of that variable conditioned by the score features. The performance encoder is only used during training. For performance generation, the performance style vector is randomly sampled according to the learned mean and variance parameters and fed to the performance decoder. Alternatively, the encoder can be used in a reference performance to extract its latent style vector so it can be used for generating performances of different pieces, in theory transferring to them the style from the reference.

Finally, the performance decoder is also a recurrent model. Two LSTMs work in parallel, for the note and beat levels, taking as inputs a concatenation of the encoded score, the performance style vector, and the outputs from previous steps in the sequence from both note and beat levels. Velocity, timing, articulation, and pedaling are all outputs produced on the note level, whereas tempo is generated on the beat level.

VirtuosoNET was also trained using performances from the MAESTRO dataset. However, only audio and corresponding MIDI files are available in it, and the authors were interested in modeling a full mapping from score to performance. To make that possible, scores for 226 pieces in the musicXML [2] format were matched and automatically aligned

[2] MusicXML 3.1 Specification. The W3C Music Notation Community Group, 2017. URL:

to 1,052 piano performances from the original dataset, making it the largest dataset of its kind at the time. Evaluation for the system was conducted first by comparative analysis of the mean squared errors (MSE) in performance predictions and later perceptually by listeners. The mathematical analysis adds credibility to the proposed hierarchical architecture, as it shows superior predictive capacity to a simple multi-layered LSTM encoder. Likewise, the exclusion of the measure level in score encoding also negatively impacts the same metric. Perceptual analysis by five pianists again favors the final architecture, even over the Basis-Mixer (Cancino-Chacón and Grachten, 2016) – an available implementation of the models by Cancino-Chacón and Grachten.

A later paper by Jeong et al. (2019b) proposes modifications to VirtuosoNET, encoding the score at the note level by means of a gated graph neural network (GGNN) [3] that structurally represents several relationships between notes, such as slurs and voices. Evaluation in the same terms shows modest improvements, especially for generated tempo curves.

As of this writing, the application of deep-learning architectures to creative and artistic tasks has been spreading in a fast pace, and with the development of each new modeling design we see a surge of reports of their results in various tasks, including expressive performance generation systems. In spite of the prolific output, it remains challenging to navigate this research field due to the variety of scopes and requirement specificities imposed by the proposed systems, making them largely incomparable and often impossible to reproduce. In fact, the success of most models is defined in equal measures by the maturity of its documentation as by the quality of its generated performances. We hope that the works described in this chapter can provide justification to the thought process applied to our own design choices, and put our contributions into perspective.

2.2 Technology-Enhanced Learning of Musical Expression

Technology-enhanced music learning is an extensive and active research field that has found its way to many classrooms and commercial applications, and would be far too complex to cover here. We merely present some of the works that seek to facilitate the

https://www.w3.org/2017/12/musicxml31/.

[3] for details, see Li et al. (2017).

learning of musical expression, be it simply by delivering information about the expressive content of performances in the form of visualizations or reports, or by evaluating a complete pedagogical approach to this subject.

The pianoFORTE (Smoliar et al., 1995) is an early example of computer system that, by taking advantage of sensory capabilities of MIDI keyboards, plots information about timing, dynamics, articulation, and voice synchronization of a piano performance with the goal of facilitating communication between music educators and students about expression. Because the system lacks a model of expression, the knowledge of the music teacher remains as the only source for feedback about a performance, so the experiment stands out as a very good example of application scenario for a CSEMP as investigated in this thesis.

A similar approach is also found in the InTune software (Lim and Raphael, 2010), though here, the application is assisting the visualization and control of intonation for wind, brass, and stringed instruments as well as voice. Though intonation can be used for expression, the study explores only a simpler application, differentiating correct and incorrect pitch intonation. In a user evaluation, the authors were able to confirm that InTune helped the majority of participants identify intonation errors they wouldn't have otherwise noticed. They also conduct a simple usability survey, from which it is clear that most users are more comfortable using score and pitch views to a spectrogram for this task.

Some models of performance for violin instruction have been proposed, though their focus lie in predicting proper posture and bowing motion rather than expression. We highlight the CyberViolin (Peiper et al., 2003) for articulation classification and the MusicJacket (Van Der Linden et al., 2011) for posture correction via vibrotactile feedback. The CyberViolin system uses electromagnetic motion tracking to classify in real-time the type of articulations played by a musician. In their envisioned application scenario, the correct articulation is known for each note, and the virtual tutor is in charge of verifying if students articulate the proper technique in each case. The MusicJacket system explores new forms of providing performance training assistance in an experiment where students improve their bowing skills assisted by a wearable device with vibrotactile feedback capabilities. However, in any case like the above, the system must have a prior knowledge of a ground truth, that is, a reference to form the basis of its feedback.

A few case studies of a complete experience teaching expression with the assistance of technological tools do exist. Hamond et al. (2019) report on the results of one-to-one piano lessons of an advanced student augmented by the use of a digital audio workstation (DAW) connected to an electronic piano via musical instrument digital interface (MIDI). The built-in functionalities of the DAW provided both real-time and post-hoc visual and auditory feedback of the performances, the visual feedback being in the form of a piano roll view with notes color-coded by velocity. Analysis of the video records and interviews with both teacher and student indicate that they viewed all functionalities positively, but seemed to rely more on post-hoc analysis. Particularly, the visual feedback of velocity revealed that when playing along with the teacher, the student had more salient articulations in the left hand than when playing alone, which prompted them to practice that section emulating their previous, accompanied attempt.

Sadakata et al. (2008) provide a very informative analysis of the impact of real-time visual feedback to the learning of expressive rhythmic skills. In the evaluation setup, participants were instructed to imitate a reference performance assisted by a tool that presented a graphical representation of its loudness and timing variations. The study also included an analysis of transfer of learning, in which previously unseen rhythms had to be performed during the test phase. Learning success was measured in terms of reductions in RMS errors between pre and post tests on each expressive dimension when comparing participant and target performances. Under this criterion, the proposed visualization enhanced only the learning of loudness, since the group that received visual feedback improved more than the control group in that aspect, but not in terms of timing. The experimental setup from this study stands out for its clarity and methodical organization, and informs our own methods in the analysis presented in chapter 6, though its strictly quantitative approach is a better fit for mature solutions, where a strong endorsement or rejection, rather than nuanced opinions, is the ideal outcome.

An experiment described by Juslin et al. (2006) presents the most advanced attempt in the literature of providing feedback about expression based on conclusions drawn from a computer model. In their study, jazz/rock guitar players were asked to play a piece expressing a certain emotion, such as sadness, and given feedback by a computer program about specific changes in expression, such as playing slower or more *legato*, in order to better approximate the model's predictions for that piece and mood. The

performances after the feedback were contrasted with those of other groups that received feedback from a professor or no feedback at all in a listening experiment, showing that the computer suggestions were the most effective in helping players convey the intended emotions.

A surprising observation is that, when inquired about it, music students often manifest resistance to the use of technological tools with pedagogical purposes (Lindström et al., 2003b; Juslin et al., 2006; Karlsson et al., 2009). However, the concrete approaches discussed here have been almost universally well-received. One logical explanation could be that there is an inherent bias in favor of technology by people who volunteer themselves to participating such pioneer studies, but it might also be the case that many musicians are still unaware of the benefits that can be reaped from a technologically-enhanced music classroom.

Chapter 3

Performance Modeling by Phrase Similarity

3.1 Introduction

This chapter describes the development and evaluation of a data-driven system for expressive performance modeling applicable to violin melodies and robust to training with a reduced number of reference performances. The ability to apply generated outputs as creative suggestions to students in the process of learning and practicing expressive performance is a central guiding principle in its design.

The proposed model uses phrases rather than single notes as units of analysis using the following approach: each phrase in a target score is matched to similar phrases from performances by experts, adapting the experts' expressive features to render a performance of the target score. This approach shares benefits that were highlighted in our discussion of the DISTALL system (Tobudic and Widmer, 2006), due to the instance-based nature of both designs. By reusing the expressive content found in a human performance, the model passively preserves much of the coherence among expressive parameters and their relationship with the melody, ideally eliciting some form of "musical ELIZA effect"(Hofstadter, 1995) on listeners. The model's creations gain more legitimacy if one sees musical performance as being composed of *idioms*, much like human languages [1] –

[1] a view shared to some degree by many music theorists, from Cooke (1989) to Zbikowski (2017).

if we consider any simple communication such as, say, ordering a cup of coffee, all languages offer a multitude of ways to achieve this, but a visit to a café quickly reveals that not all of them are equally likely to be heard. In the musical context, an instance-based model hopes to capture elements of performance conventions, the recognizable ways in which phrases, consciously or not, are played.

modeling idioms of performance fits the educational purpose of our system as well, ensuring that music students are stimulated to learn performance patterns that align with the expectations of an audience of the target genre, giving them the necessary awareness to incorporate or break from these conventions at will.

The chosen approach belongs to a class of methods called *lazy learning*. Systems of this kind defer analysis of the dataset until the time of evaluation (Aha, 2013) (for us, until performance generation). This gives the system flexibility to alter the dataset at will at the expense of slower execution, an opportunity for creative uses of musical references according to the section of the performance to generate.

To assess the ability of the proposed modeling framework, we take advantage of a small corpus of multimodal recordings and apply it to the generation of signals related to motion, specifically, bow velocity during violin performances.

The results from the work described in this chapter have been published in the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education Ortega et al. (2017a).

3.2 Method

The proposed performance generation process can be divided in three parts, as illustrated in figure 3.1: first, the musical score to be performed is segmented into a sequence of phrases. Then, for each phrase, the system searches the performance database for the most similar phrases available according to a similarity heuristic that we defined. Finally, the performance from the most similar phrase is processed and incorporated as the newly generated performance. In that sense, the model can be seen as an application of nearest-neighbor regression.

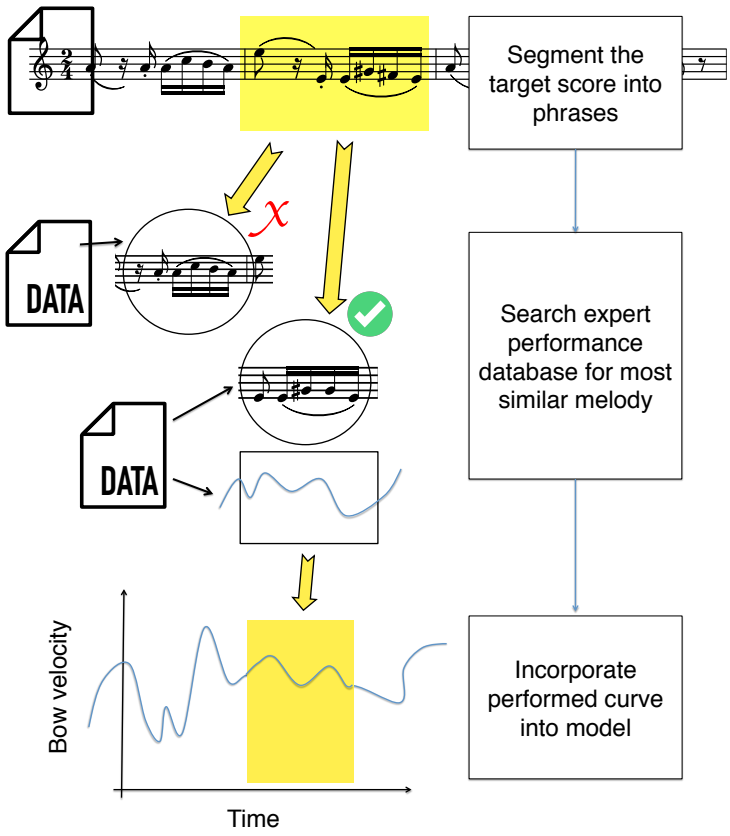


Figure 3.1: Outline of the performance generation method

In the pedagogical scenario we envision, the training set can be processed *a priori*, and that allows for more human interference and manually crafted features. However, in the interest of being able to modify the training set composition on the fly as well as increasing the method's reproducibility, we refrain from doing so, and instead adopt a fully automated process for the training set preparation, which we will discuss alongside each of the above three steps from the generation.

In the first step of the performance generation process, the goal is to divide pieces into their constituent phrases, which are the actual objects of analysis. Because we are in-

terested in fully automated performance generation, we opt for a standardized phrase size of two measures. Naturally, the duration and number of notes in a two-measure segment can vary widely from one musical piece to another, but for the purpose of establishing a fixed phrase size, the two-measure mark is a logical decision, as it is typically long enough to encompass a complete musical idea and short enough to allow the perception of gradual change in interpretation.

The second and crucial step in our performance generation method is identifying the best matching phrases within the dataset of known performances for every phrase of the desired piece. This requires defining a measure of phrase similarity that translates our intuitive perception of the concept with the fewest possible aberrations. Here we introduce a few simplifications: given that we opted to evaluate the method on solo violin performances of western classical music, it makes sense to compute similarity based solely on note pitches and durations of monophonic melodies. Whenever we encounter notes that are played simultaneously (double-stops), only the highest pitch is considered part of the melody. With this simpler formulation, we take advantage of a measure of melodic dissimilarity proposed by Stammen and Pennycook (1993), which is an adaptation of the dynamic time warping (DTW) algorithm.

In this melodic dissimilarity algorithm, melodies are encoded as pairs of sequences, one sequence representing tempo-invariant note durations, and the other, key-invariant note pitches. When comparing melodies, the minimum necessary warping cost to turn each sequence from one melody into the corresponding sequence of the other melody is computed using DTW, and the total measure of dissimilarity is taken to be the sum of squares of the warping cost of each sequence. For more details on the DTW algorithm itself, we refer the reader to Müller (2007).

The third and final step for performance generation is the adaptation of performance data from the reference phrase selected in the previous step. Here, the most adequate adaptation method may vary according to the desired output feature. For the evaluation of the model, we attempt to predict violin bow motion information for each note, separated into two features: mean absolute bow velocities and bow motion direction.

Figure 3.2 breaks down the adaptation process as implemented for evaluation of bow velocity prediction. The graph represents absolute bow velocity information from the

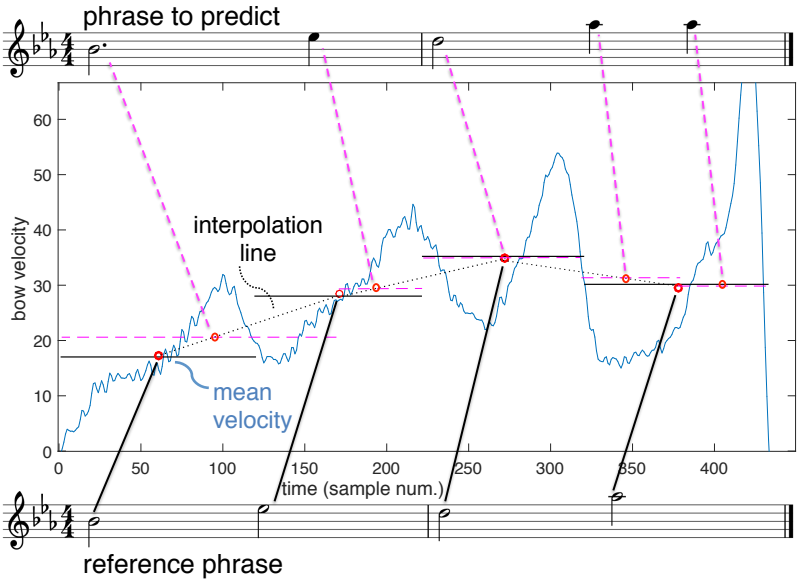


Figure 3.2: Adaptation of bow velocity data from a reference phrase.

reference phrase seen in the musical score below it. Because of the articulation of each note in the phrase, a certain degree of variation in bow velocity is observed along the duration of each note. Our method tries to eliminate the influence of such variation computing the mean bow velocity value for the performed duration of each note (indicated by horizontal lines), because that value is indicative of the dynamics of the performance. The dotted line tagged "interpolation line" is an approximation of the variation in mean bow velocity along the duration of the whole phrase. Since the phrase to be predicted can differ rhythmically from the reference, the mean bow velocity values predicted for each target note are sampled from the interpolation line in the instants corresponding to the center of each note's duration. In the example from the figure, mean velocity increased in the reference phrase between the first and second notes; so, since the duration of the first note in the phrase to predict is longer than the first note of the reference, its predicted mean velocity is also higher. The third note of both phrases, on the other hand, has the same duration and relative placement within the phrase, so the predicted

mean velocity matches the reference exactly.

The adaptation process for bow direction prediction is simpler: each reference note is labeled as either bow down or bow up according to its onset velocity. Directions of notes to predict assume the same label as the closest note onset in the nearest-neighbor reference for the normalized moment of its own onset.

All performance data used in the evaluation were collected for prior experiments on ensemble expressive performance (Marchini et al., 2014; Papiotis et al., 2014). This dataset, known as the Ensemble Expressive Performance (EEP) dataset, consists of recordings of the String Quartet no. 4, Op. 18 by Ludwig van Beethoven, organized in smaller sections and performed in various conditions. We selected excerpts exclusively from the first violin. Phrases which repeat themselves throughout the piece are included only once to avoid positively biasing the nearest-neighbor algorithm. Bowing motion data were acquired by means of a Polhemus Liberty wired motion capture system as detailed by Maestre (2009), at a sample rate of 240 Hz. Performance audio was recorded from a piezoelectric pickup attached to the bridge of the instrument, and timestamps for note onsets and offsets were manually annotated. Synchronization of audio and motion capture data was reviewed through the alignment of linear timecode timestamps. The scores for symbolic data processing were input in MusicXML format and all code was written in MATLAB.

As previously mentioned, efficacy of the model was evaluated in terms of its ability to predict expressive signals from the chosen dataset. A leave-one-phrase-out process was adopted, so that bowing motion information was predicted for every phrase in the dataset using the remaining phrases as references.

3.3 Results

A total of 68.75% of notes had their bowing direction correctly classified. In contrast, the baseline most frequent label classifier shows an accuracy of 50.94%. Though it seems possible to improve accuracy with simple modifications to this model, we consider bowing direction classification simply as a quality indicator for the method, and instead give more attention to the analysis of bow velocity magnitude predictions, since that is a

feature determined more by stylistic choice and less by physical restrictions of the instrument when compared to bowing direction.

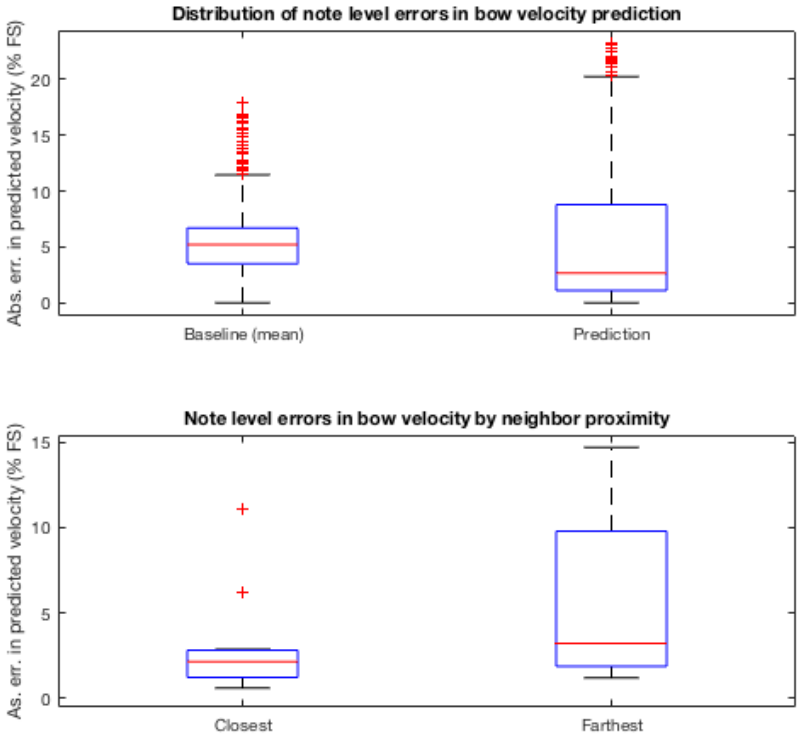


Figure 3.3: Boxplot of prediction errors in a leave-one-phrase-out approach.

The top plot in figure 3.3 shows the distribution of absolute errors per note in predictions as a percentage of the full-scale value. The baseline model is the prediction of constant velocity equal to the mean absolute velocity value. We can see that the model achieves lower median error than baseline at the cost of a wider error distribution profile. Since the dataset is very small, we are unable to reject the null hypothesis of equality between baseline and prediction means in a one-sided t-test. Instead, in order to validate our measure of phrase similarity as a predictor for bow velocity, we performed a different test. We partitioned all predicted phrases into two sets of equal size classified by melodic dissimilarity score, that is, one set containing phrases for which we could find

the *closest* nearest neighbors during prediction, and the other, phrases for which the nearest neighbor in the reference set were the most distant. The bottom plot in figure 3.3 shows the distribution of errors for these two classes. In this case, t-testing confirms that phrases with closest neighbors present the lower mean absolute error of the two sets with $p = 0.01$.

We also observe how the model compares to a constant velocity baseline on the phrase level. Figure 3.4 shows how the expected mean velocity along entire phrases for the modeled performance is a better approximation than baseline. Since the absolute bow velocity curve and the loudness curve extracted from the audio recording share a high coefficient of determination ($R^2 = 0.76$), this could be interpreted as evidence that the melodic content of a phrase is indicative of its character in dynamics, i.e.: if a given melody is typically played *forte*, another, similar melody, should be played *forte* as well.

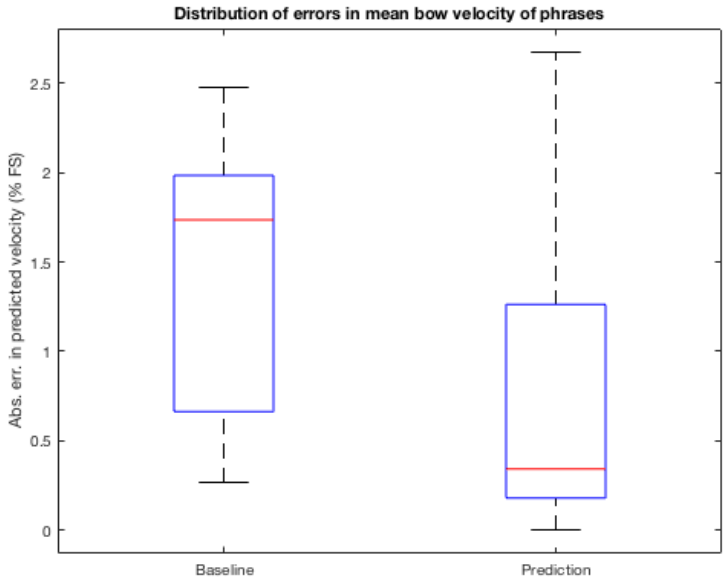


Figure 3.4: Boxplot of errors in mean phrase velocity predictions.

3.4 Discussion

Overall, the violin bow motion predictions obtained from the phrase similarity model were successful in validating the hypothesis that performances of similar melodies share expressive traits. The observed reduction in prediction errors compared to a baseline prediction also indicate that the specific methodology adopted for processing musical information is adequate for this purpose. In particular, we highlight the phrase similarity heuristics and the data conditioning for expressive feature adaptation, which demonstrate potential to be reused in models of higher complexity.

Naturally, it should be noted that the evaluated settings are very specific in terms of musical style, predicted features, and most of all dataset size, which means that little is known about the behavior of the model in other, more general conditions. The evaluation using phrases extracted from the same piece and played by the same performer may have led to a positive bias in the results. On the other hand, using data from a variety of pieces and performers could have had a much stronger, negative bias, failing to provide examples of melodies that share melodic traits and that could allow us to verify a shared vocabulary of performance features. By separating predictions made from similar melodic content from those predicted using contrasting melodies, we sought to mitigate any biased effect and to understand the potential of the model given a properly catered dataset of reference performances, which would require a long-term effort in data collection and preparation.

The improvement in bowing direction predictions over baseline is a surprisingly positive result, given the relative simplicity of the model's input features. It is notoriously difficult, even for musicians themselves, to anticipate the best bowing strategy for an unknown piece, but the results hint that, for a fair amount of notes, the rhythmic and melodic patterns are sufficient information to accurately predict the best bowing direction. Though this is not likely the best computational strategy for this task, it serves as evidence that the proposed similarity measure carries some musical meaning.

The analysis of bowing velocity was our main goal, since it has direct implications for expressive performance teaching. Confirming that a phrase-level approach is more accurate than trivial baselines in predicting expressive features is simply a first step towards proposing computer systems that can contribute meaningfully to the creative process

behind developing an interpretation of a musical piece. The generated velocity signals, when interpreted as predictions of a reference performance, exhibit larger variance in the error distribution than the variance of velocities in the piece itself, which indicates that although the mean error values are lower than baseline, the generated velocity profile of some phrases conflicts with the one observed in the reference performance. This is not a surprising discovery given the nature of the problem. One fact that partially justifies that finding is that there are multiple possible interpretations for any given musical excerpt, which means, for example, that something as simple as using a recording from a different musician as reference could yield very different error profiles while being just as acceptable perceptually. Another decisive fact that helps interpret this result is a significant design limitation of this model: because phrase similarity is the single characteristic used for constructing the output signal, if the training database is kept equal for the entire generation process, repetitions of a phrase will all produce identical performances. This is typically undesirable as it is an unrealistic behavior when contrasted with a real musician. Likewise, the lack of contextual information in the performance generation can produce abrupt changes from phrase to phrase which are not evident in the numerical analyses presented, but that could also feel unnatural to a listener.

In conclusion, despite the simplicity of the model and the limitations we discussed, the proposed modeling design shows robustness in face of very small training datasets, and for the same reason offers versatility, since clever manipulation of the dataset during generation can completely change the character of generated performances. It also offers a meaningful contribution to the field, that still lacks body of knowledge in the development of non-piano, fully-automated CSEMP, particularly with an emphasis on phrase-level analysis.

Chapter 4

Performance Modeling by Phrase-Level Feature Parameterization

4.1 Introduction

The work described in this chapter can be seen as a natural continuation of the model development described in chapter 3. As highlighted in the discussion of that system, the proposed instance-based performance generation approach is tolerant of small-scale datasets but still degrades in quality in the absence of sufficiently similar reference phrases. The method also lacks an ability to take contextual information of musical phrases into account, which can be an important asset for preventing the generation of repetitive and dull performances. Moreover, as we move towards larger datasets for model training, and turn our focus to generating audio rather than motion features of expression, a notable limitation of the previous approach is its inability to have multiple reference samples contributing to the output, or, in other words, "blending" interpretations. With these issues in mind, a summary of the goals addressed in this chapter are:

- To improve upon our phrase-level CSEMP, with a method for EPA feature parameterization that lets it combine contributions from different training samples and take advantage of more powerful learning algorithms;

- To develop an automated method of phrase segmentation that can, as best as possible, reflect the musical structure of pieces;
- To validate the resulting system as a generator of expressive features present in audio recordings, such as dynamics and timing variations; and
- To contrast the efficacy of phrase-level modeling against note-level modeling under settings relevant to our scenario.

The audio feature most deeply analyzed in these experiments is loudness. Musicians actively manipulate loudness in music performances, and the variations they introduce as an expressive resource form what is called *musical dynamics*. However, because modeling requires quantifying these phenomena, it is important to establish a distinction between musical dynamics itself as a perceptual feature – as well as the actions made by musicians to create it – and the loudness of the audio in a recorded performance. It is logical to conclude that by virtue of the recording process, several elements external to the performance impact the loudness of the resulting audio signal, such as the sensitivity of the microphones, their distance to the sound source, compression and other post-production processing techniques used in mixing and mastering, and so on. Also an issue that arises is mapping loudness as a physical quantity related to sound pressure to loudness perceived by the human ear, since the latter is frequency-dependent and the relationship between the two is known to be non-linear and complex. Lastly, it should be noted that the majority of traditional musical instruments vary in timbre according to the intensity with which they are played. As a result, dynamics are perceived not only as modulations in volume, but also as changes in sound character of great importance to communicate a performer's expression. For an overview of difficulties relating loudness to musical dynamics, see (Patterson, 1974).

Unsurprisingly, in light of all factors mentioned above, most data-driven CSEMP that concern themselves with the generation of dynamics are based around some feature of the musical instrument itself rather than the resulting loudness of the performance. For piano, the main feature for this purpose is the velocity of the hammers that strike the strings [1], made easy to measure by the fact that velocity sensors have become ubiquitous in electronic pianos. The experiment presented in chapter 3 is another example, as

[1] A detailed analysis was published by Repp (1993).

the main quantity associated with loudness in violin performances is the velocity of the bow; nevertheless, in the violin like in many instruments, every playing style choice affect multiple sound features at once [2], so other features such as the contact point between bow and string and the tilt angle of the bow also impact dynamics in some degree.

Despite these challenges, researching effective methods to utilize recording loudness information as a source for performance dynamics generation can facilitate modeling a wider variety of instruments without needing intrusive sensors or specific data collection, ultimately helping us learn about music expression in a broader sense.

Results from this work have been published in the 10th International Workshop on Machine Learning and Music (Ortega et al., 2017b), in Frontiers in Psychology (Ortega et al., 2019a) and in the 12th International Workshop on Machine Learning and Music (Ortega et al., 2019b).

4.2 Method

In the process of achieving the previously outlined goals, a series of experiments were conducted, each with specific purposes. We present their shared methodology below, highlighting differences when applicable.

Model enhancements

Recalling the structure of our CSEMP as shown in figure 3.1, the first step, both in training and generation, is to determine phrase boundaries to segment musical pieces into the units of analysis.

Considering that an important function of music interpretation is to highlight its structure (Palmer, 1997), if the piece segmentation method is set to follow musically meaningful boundaries, the generated performances may also fulfill that function, and sound more natural as a consequence. The same is desirable when processing the training set to obtain the reference phrases which will be used in generation. A musically meaningful segmentation of the training set pieces ensures that features extracted from the resulting phrases are more likely to capture complete ideas which would adequately translate to

[2] a desirable characteristic in musical instrument design, according to Jordà (2004).

a similar phrase. Under this premise, we abandon the fixed length of two measures in exchange for a dynamically-sized but fully automated approach.

Our piece segmentation algorithm is built upon the local boundary detection model, or LDBM (Cambouropoulos, 2001). The LBDM is a method that attributes a score between 0 and 1 for each note in a piece which represents the likelihood of that note being a local boundary (i.e.: the first in a phrase) taking into account the variations in pitch, note durations and presence of rests. Our model uses an implementation of it available in the MIDI Toolbox for MATLAB (Eerola and Toiviainen, 2004). The segmentation algorithm uses the method's score values for recursively evaluating whether a segment should be split further. The following pseudocode illustrates how that is achieved:

Algorithm 1 Piece Segmentation

Input: The LBDM value l_i for every note $i = 1..end$.

procedure SEGMENT($[l_k, l_{k+1}, ..., l_n]$)
  if $n - k \leq 10$ **then**
    return one segment, from k to n
  else
    Calculate z-scores from values $[l_{k+2}, ..., l_{n-1}]$
    if the largest z-score $z(l_{max}) > 2$ **then**
      return `Segment`$([l_k, ..., l_{max-1}])$, `Segment`$([l_{max}, ..., l_n])$
    else
      return one segment, from k to n

As a result of its structure, the algorithm gravitates towards phrases of approximately 10 notes without imposing a hard restriction. Even though phrases of a single note might be musicologically acceptable, we intentionally prevent their occurrence, since pieces with ambiguous phrase boundaries often cause the LBDM to output high likelihood values for consecutive notes in situations where one-note phrases would not be reasonable.

Even though the analysis is done on a phrase-level, the model is designed to generate expressive performance output features on a note-level. The chosen features for the experiments are the mean level of loudness for each note, their onset time and duration. The mean loudness level L_n of each note is computed according to equation 4.1, where s are audio samples in the time-frame of note n.

$$L_n = 20 \cdot \log \left(\sqrt{\sum_{i=\text{onset}_n}^{\text{offset}_n} s_i^2} \right) \tag{4.1}$$

This process leads to loudness values measures in decibels relative to full-scale (dBFS). When synthesizing performances generated by the model, these values are converted into the MIDI velocity scale. This conversion follows the findings of Dannenberg (2006), who observed a square-law between velocity values and RMS amplitude in synthesized audio. Our mapping was empirically calibrated by synthesizing performances from our recording database and ensuring that the dynamic range of the synthesized audio matched that of the original recording.

The next important enhancement to our model design is the development of a method of performance loudness signal parameterization. This process allows us to associate each phrase in the training database with a set of numeric descriptors that summarize the character of that performance's loudness signal. These descriptors also have important properties for the purpose of performance generation. The first one is that they are *normalized* with respect to the scope of the whole musical piece, leading them to describe only the local character of the phrase and making adaptations to a different context simple. The second relevant property is the smoothness of the loudness function with respect to the descriptors, which ensures that value variations only gradually modify the character of the reconstructed signal, increasing the robustness of the representation and simplifying its direct numerical manipulation.

Figure 4.1 shows a loudness curve plot for a piece section from the dataset, discretized on the note level. The loudness at each note n is L_n, as computed according to equation 4.1. Between dashed lines is the section of a particular phrase in that piece. M is the mean level of the piece, whereas m is the mean level of the phrase. The dynamic range is given by R for the piece and r for the phrase. Considering that pieces may be performed at widely different mean levels and dynamic ranges, if we intend to use phrases from multiple pieces as references for prediction, it makes sense to measure their values relative to M and R and allow these to be set by the user for the predicted rendition. When analyzing pieces from the training set, these metrics are measured as follows, where d_i represents the duration of a note i:

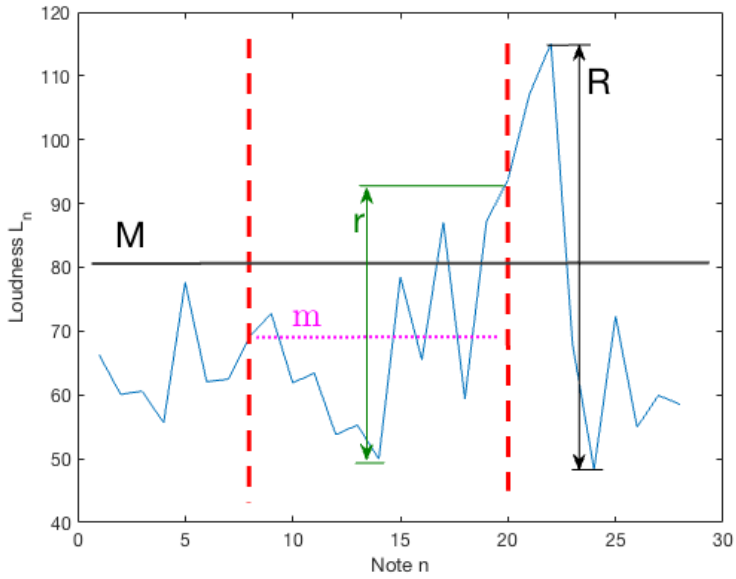


Figure 4.1: Performed loudness for a section of a piece and some key measurements.

$$M = \frac{\sum_{i=1}^{n} L_i \cdot d_i}{\sum_{i=1}^{n} d_i} \qquad R = \max_i(L_i) - \min_i(L_i) \tag{4.2}$$

Essentially, M is the weighted mean of loudness values L_i with respect to the durations d_i, and R is the range of excursion of the same random variable.

Using these metrics, we define the mean loudness level m_p of a phrase p having w notes and beginning in note k as:

$$m_p = \frac{1}{R} \cdot \left(\frac{\sum_{i=k}^{k+w} L_i \cdot d_i}{\sum_{i=k}^{k+w} d_i} - M \right) \tag{4.3}$$

This descriptor (m_p) refers to the character of a phrase (e.g.: *forte*, pianissimo, *etc.*) and

is represented by a single real number that measures how the loudness in the phrase deviates from the mean loudness in the piece, normalized by that piece's dynamic range.

The second descriptor is the phrase dynamic range r_p:

$$r_p = \frac{1}{R} \cdot (\max_p(L_p) - \min_p(L_p)) \tag{4.4}$$

Analogously to what happens between m_p and M, the descriptor r_p is the phrase-level counterpart of R, and it measures the dynamic range of a phrase in "piece range" units.

Lastly, a third descriptor is the phrase contour (C_p), a function which describes how each note in a phrase p contributes to its loudness once the effects of m_p and r_p are discounted. Therefore, if m_p, r_p, and C_p are known for p, the loudness value for each note $i \in p$ can be determined by:

$$L_i = M + R \cdot (m_p + r_p \cdot C_p(i)) \tag{4.5}$$

This definition has two implications. The first is that we can determine the values of C_p in each note i for phrases in the training set since their loudness values L_i are known. Using those values, C_p is fitted in a quadratic polynomial using the least-squares method, as inspired by observations regarding typical phrasing dynamics by McAngus Todd (1992) and other researchers (Gabrielsson et al., 1983; Tobudic and Widmer, 2003; Livingstone et al., 2010) who all point to their parabolic contour. This allows us to parameterize C_p for all p in the training set using only three coefficients for each.

The second implication of equation 4.5 is that determining m_p, r_p, and C_p for a phrase p for which we wish to generate dynamics suggestions is enough to compute suggested loudness values L_i for all notes $i \in p$, as long as chosen values for M and R are provided. In practice, M represents the overall character of the piece and R its overall dynamic range, to which all phrases should conform. These can be set to default values or adapted according to a wider context (e.g.: a lower M for an *adagio* than for an *allegro*). This aligns with our initial desire of characterizing phrases independently of context, so that the knowledge-base of the training set is applicable across all musical intentions.

As previously mentioned, the proposed modeling is meant to output suggested mean levels of loudness for each note but also a couple of timing-related features: note onset times and note durations. By including features of timing besides dynamics we achieve a minimal set of performance features that is sufficient to render synthesized performances that can be evaluated perceptually. If timing features are omitted, the resulting performances sound noticeably uncanny, and hinder our ability to evaluate the quality of the generated loudness suggestions.

As was the case with loudness, some parameterization of timing features is needed to make the best use of the phrase-level analysis. The local tempo curve is defined as the function that describes how the tempo changes throughout the phrase. For each note i, its local tempo value t_i is computed as:

$$t_i = \frac{60}{T} \cdot \frac{b_i}{\text{ioi}_i} \tag{4.6}$$

Where T is the piece tempo in beats per minute, b_i is the duration of note i in beats according to its rhythmic figure in the score, and ioi_i is the inter-onset interval between notes i and $i + 1$. The local tempo curve of each phrase is, once again, the quadratic polynomial that best fits its local tempo values, for x–axis values spaced proportionally to b_i.

For a suggested local tempo curve τ, one can use equation 4.6 to compute the IOI of each note, since t_i is given by $\tau(b_i)$ and the desired piece tempo T should be provided. Assuming that the first note starts at os and working sequentially, this defines the onset times of all notes.

ANN model

The previous model based on phrase similarity was conceived with the purpose of being a good fit for small-sized datasets. Despite that, it is relevant to explore how phrase-level modeling with the proposed segmentation and parameterization methods can take advantage of a larger number of reference performances. As the dataset size increases, running a DTW algorithm against all phrases for each new phrase performance generation becomes too time consuming. Rather than seeking optimization alternatives for

the same algorithm, we test the efficacy of a phrase-level analysis in a performance prediction task using the same piece segmentation algorithm and output feature parameterization but replacing the phrase similarity heuristic with a set of phrase features in a feed-forward artificial neural network (ANN) learning algorithm. This choice is justified by ANNs' ability to process very large amounts of data as well as its high modeling power and good generalization results as shown by systems presented in chapter 2.

Table 4.1 summarizes the input features used for training. Piece keys and modes were estimated from pitch profiles as detailed in Temperley (1999). Parameterization of phrase loudness signals was slightly modified: loudness measurements of each piece were computed in windows of 0.1s following the EBU R 128 standard (EBU TC Committee, 2016) normalized to zero mean and unit variance to eliminate differences caused by inconsistent recording conditions. This method of calculation provides some correction to account for the physiological perception of loudness while maintaining methodological reproducibility. The loudness curve of each phrase is represented simply by the three coefficients of the quadratic polynomial that best approximates it under a normalized duration between values 0 and 1. This summarizes the previously defined parameters into only three numbers, at the cost of some clarity in the interpretation of parameter values. Having more parameters per phrase was not an issue for k-nearest neighbors (k-NN) learning, but may impact performance in an ANN model.

To facilitate the optimization task, some data conditioning was performed. Phrases with less than 4 notes and outliers (z-score above 10 in any feature) were discarded, all nominal features were converted to "one-hot" format and all numeric features were standardized. The processed dataset was divided into training and test sets containing 90% and 10% of instances, respectively.

The feed-forward network was programmed in the PyTorch[3] framework and built with two hidden layers of 25 nodes each, using ReLU as an activation function and standard mean-squared error as a loss function. The training was run for 1800 epochs in stochastic gradient descent optimization with batches of 100 instances, learning rate of 0.2 and momentum of 0.1. The learning rate was decreased by a factor of 10 every 600 epochs. All parameters were cross-validated using a subdivision of the training set prior to the final training round.

[3] http://pytorch.org

Feature	Data type	Description
Beat in Measure	$x \in [0, 4]$	The beat where the phrase begins.
Metric strength	$x \in \{3, 2, 1, 0\}$	How strong the start beat is. e.g.: down beat $= 3$.
Number of notes	$x \in \mathbf{N}$	Total of notes in phrase.
Duration	$x \in [0, \infty)$	Phrase duration in beats.
Location in piece	$x \in [0, 1]$	Where in the piece the phrase is played.
Pitch curve coefficients	$x_0, x_1, x_2 \in \mathbf{R}$	Quadratic coefficients approximating the MIDI pitches of phrase notes.
Pitch contour coefficients	$x_0, x_1, x_2 \in \mathbf{R}$	Quadratic coefficients approximating the variation in MIDI pitches of phrase notes.
Rhythm Drops	boolean	Whether a note with higher duration follows another with shorter duration in the phrase.
Rhythm Rises	boolean	Whether a note with shorter duration follows another with higher duration in the phrase.
Rhythm contour coefficients	$x_0, x_1, x_2 \in \mathbf{R}$	Quadratic coefficients approximating the variation in duration of phrase notes.
Strongest note location	$x \in [0, 1]$	Where in the phrase is the note with highest metric strength.
Piece key	A - G#	Tonality estimation of piece.
Piece mode	Major/Minor	Mode estimation of piece.
Chord probabilities	$x_0..x_6 \in [0, 1]$	Estimated diatonic chords presence probabilities.
Initial chord degree	I - VII	Most likely chord in phrase start.
Final chord degree	I - VII	Most likely chord in phrase end.
Has Dissonance	boolean	Whether there are notes from a different tonality.
Dissonance Location	$x \in [0, 1]$	Location of first occurrence of dissonant note.
Is solo piece	boolean	Solo or ensemble piece.

Table 4.1: Input features of the model.

Data collection

In order to augment the EEP dataset used in the previous experiment, we conducted recordings of a professional violinist, generating a new dataset, which we call *Expressive Solo Violin Dataset*, or ESV [4].

Eight short (approximately 50s each) musical excerpts were recorded to be used for both model generation and evaluation. The pieces were chosen from the violinist's repertoire with the intention of providing a wide range of moods and melodies of western classical violin, and were played solo and without metronome. Each excerpt was performed three times with different directions: once as inexpressively as possible, once as the musician believes they would normally play, and a third time exaggerating all expressive actions. Only the exaggerated versions were used in the evaluation of our CSEMP, but we believe the exercise, besides providing these contrasting interpretations for future research, was helpful for raising the musician's awareness of their own expression and thus compensating for the lack of a concert atmosphere that would put them in the necessary mental state for a convincingly expressive performance. The audio was captured with multiple condenser microphones, one placed at close distance from the violin body, another placed higher, above the musician's head, and a stereo pair about 1m away from the player in an X/Y configuration. For the modeling, signals from the close ranged microphone were used for the lower level of reverberation which improved the semi-automatic audio-to-score alignment. The scores of all recorded excerpts were manually transcribed into MusicXML [5] format using the MuseScore software [6]. To compute audio-to-score alignment, the audio files were normalized to -0.1 dB and then input in the Tony software (Mauch et al., 2015), where the onsets and offsets played were recognized by a hidden Markov model designed by Mauch et al. over the pYin (Mauch and Dixon, 2014) algorithm and manually corrected. This information about onsets and offsets could then be exported to a table and their corresponding pitches matched against the pitches from score notes using an implementation of the Needleman–Wunsch algorithm for optimal matching of sequences (Needleman and Wunsch, 1970), thus remedying discrepancies between score and performance (e.g.: grace notes).

[4] Available at https://zenodo.org/record/5765676

[5] MusicXML 3.1 Specification. The W3C Music Notation Community Group, 2017. URL: https://www.w3.org/2017/12/musicxml31/.

[6] http://musescore.org

Experimental procedures

Both predictive and perceptual analyses were conducted as a means of evaluation of the proposed modeling methods. In the predictive analyses, note loudness values generated by some variation of our CSEMP are interpreted as predictions of target performances extracted from our datasets, and measures of prediction error are used as metrics of model quality. In the perceptual analyses, synthesized audio based on generated performances as well as other baseline conditions are presented to listeners, and the quality metrics are computed based on their feedback.

For the purpose of evaluating the method of parameterization of loudness signals, a predictive analysis was conducted using leave-one-out cross-validation on the pieces of the ESV dataset in four different performance generation conditions: without parameterization, with parameterization and nearest-neighbor predictions, with parameterization and k-NN predictions, and inexpressively. For a clearer comparison with the results obtained in bow motion predictions, the same analysis was performed on the EEP dataset using 10-fold cross-validation on dataset phrases.

To observe how the proposed modeling approach fares against more conventional models that rely on note features rather than phrase features, we computed 41 note features from score information and derived musicological inferences using the union of datasets EEP and ESV, and employed the resulting feature vectors for predicting note loudness values and local tempi using various algorithms as implemented in the Weka machine learning software tool, version 3.8.3 (Frank et al., 2016).

The final predictive analysis explores whether more powerful machine-learning models powered by larger performance datasets benefit from a phrase-level modeling approach with our phrase segmentation and parameterization methods. Here, the presented ANN model is trained to predict note loudness information from violin pieces from the MusicNet dataset (Thickstun et al., 2017). This corpus of performances provides audio-to-score synchronization and facilitates the calculation of melodic and harmonic features of the ANN model thanks to the abundance of chamber music pieces with individually notated instrument parts, and was thus chosen for the task. To distinguish the main melody from harmony, violin parts were treated as melodies and all other instruments, as harmony. Only the subset of pieces which contained a violin were used,

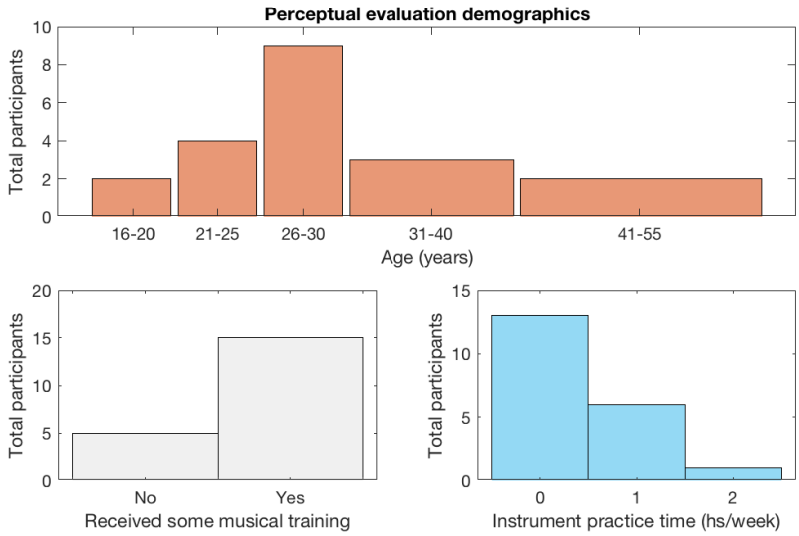


Figure 4.2: Summary of perceptual evaluation participants information.

resulting in 122 pieces and a total of 874 minutes of recordings. All presented metrics of prediction were computed using 10-fold cross-validation randomly sampled from the dataset phrases.

For the perceptual analysis, note loudness values, note onset times, and note durations were generated for all eight pieces of the ESV dataset using the nearest-neighbor implementation of the parameterized phrase-similarity model. This choice of implementation was motivated by the analysis of the first predictive experiment, as should become clearer in the following section. For the generation of each performance, the training set consisted of the remaining 7 pieces of the ESV dataset, as in a typical leave-one-out approach. The generated loudness values were converted into MIDI velocities to control the dynamics of synthesized versions of the pieces. The syntheses were made using Apple Logic Pro X's EXS24 sampler[7], with violin samples obtained from the Freesound database (Akkermans et al., 2011). The sample set was chosen for its lack of vibrato, in order to minimize the influence of this other expressive element in the evaluation of the synthesized performances [8].

[7] https://www.apple.com/lae/logic-pro/

[8] Violin samples from user ldk1609 at freesound.org, licensed with Creative Commons v.1.0

Three versions of each piece were synthesized for the evaluation, the only difference being the supplied velocity values and note onset times and durations, resulting in different dynamics and timing for each of them: one version used velocity and onset values derived from the model suggestions as described above; a second version corresponds to the expression of the performer in the recordings, as measured for usage in the training set. The third and last version serves as baseline and scientific control, and uses the same velocity for all notes, its value being the mean value used in the "human" version to minimize discrepancies in volume level, and its timing has no fluctuation, the tempo being set to the mean tempo from the "human" version. Each of the three versions of the original 8 pieces were manually divided into 3 excerpts of approximately 15s each and their audio normalized (applying the same gain to all three versions of an excerpt to prevent from modifying their relative dynamic range). Finally, the 8 most complete, melodic sounding of those 24 excerpts were selected for evaluation.

The evaluation was conducted by means of an on-line survey. Participants were instructed to hear randomized pairs of audio samples from the synthesized pieces, always consisting of two out the three existing versions of an excerpt. They were then presented with two questions for which to choose between audio samples 1 or 2: "In terms of dynamics (the intensity of volume with which notes are expressed), select which audio sample sounds most like a human performance to you." and "Which performance did you like best?". Finally, participants were instructed to answer, from 1 to 5, "How clearly do you perceive the distinction between the two audio samples?". A space for free comments was also included in each screen to encourage participants to share insights about their thought–process.

A total of 20 people participated in the experiment. Recruitment was carried out by personal invitation and each participant was assisted in accessing the web page containing the survey and its instructions using their own computers and audio equipment. Each of them was asked to provide answers to 16 pairs of melodies as described above, but early abandonment was allowed. This provided a total of 305 pairwise comparisons. Figure 4.2 shows a breakdown of the profile of participants in terms of age and musical training.

4.3 Results

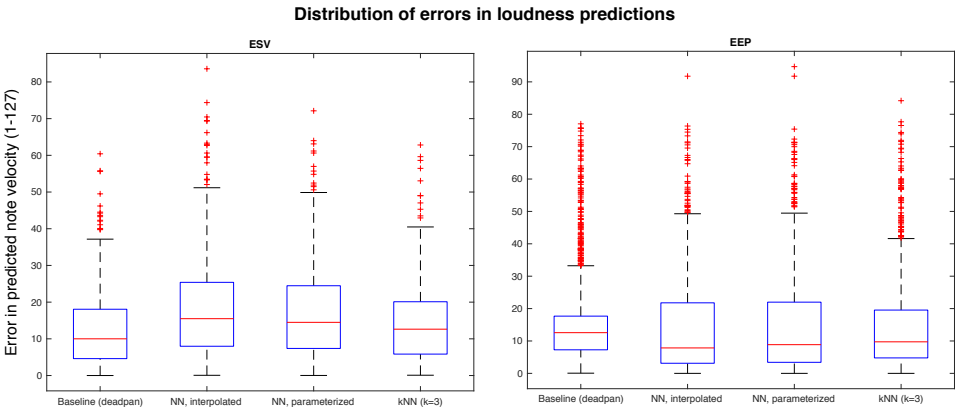


Figure 4.3: Distribution of mean absolute errors in predictions of loudness using ESV and EEP datasets.

Figure 4.3 shows our predictive analysis for the purpose of validating the proposed parameterization method in the form of boxplots of mean absolute errors in predictions of loudness measured on note level using the MIDI velocity scale. The left plot shows aggregated errors across all 8 modeled pieces in the ESV dataset, whilst the right plot presents the same metric for phrases in the EEP dataset. The baseline values measure the mean absolute difference between the dynamics of each performed note and the mean loudness of the reference performance, therefore, it represents the lowest possible errors for a prediction with no dynamic variation. The second boxplot from the left in each frame, labeled "NN, interpolated" are the result of applying a nearest-neighbor prediction with the same feature adaptation process as described in chapter 3 for bow velocity to the loudness signal, hence without using the parabolic parameterization of the phrase contour function (C_p in the model). The enhanced method, as described in this chapter, is applied to the other two measurements, the boxplots labeled "NN, parameterized" using nearest-neighbor prediction and the rightmost ones using the 3 nearest neighboring phrases in the training set. The red lines indicate the median values of each distribution.

The comparative analysis between note- and phrase-level modeling is summarized in table 4.2. As an added reference, the best deadpan performance on this dataset measures MAE of 18.02. The second and third columns indicate Pearson's correlation and mean

Algorithm	r (34 feat.)	MAE (34 feat.)	MAE (41 feat.)
SVM	0.4557	15.44	14.11
ANN	0.3789	22.71	20.08
kNN	0.5910	13.17	12.57
Random Forest	0.7319	11.18	10.49
Phrase-Level kNN (ours)	*0.2956*	*16.82*	—

Table 4.2: Performance of note–level algorithms versus proposed phrase–level method on EEP+ESV.

absolute errors obtained using only the input features related to melodic and rhythmic content of a piece, thus semantically similar to the information used in the melodic similarity calculation of our method. The rightmost column refers to errors measured after modeling with all features available, thus including score annotations such as dynamic markings, articulations, and slurs. The bottom row corresponds to the results of our method using $k = 3$. More interpretable evidence on the profile of generated performances is shown in the graph from figure 4.4, which overlays the velocity values from the reference performance of Edward Elgar's *Chanson de Matin*, opus 15, no. 1, bars 2–28 by a real violinist against the predictions from our phrase-level approach and the best-scoring note-level version of the above – achieved with a random forest algorithm.

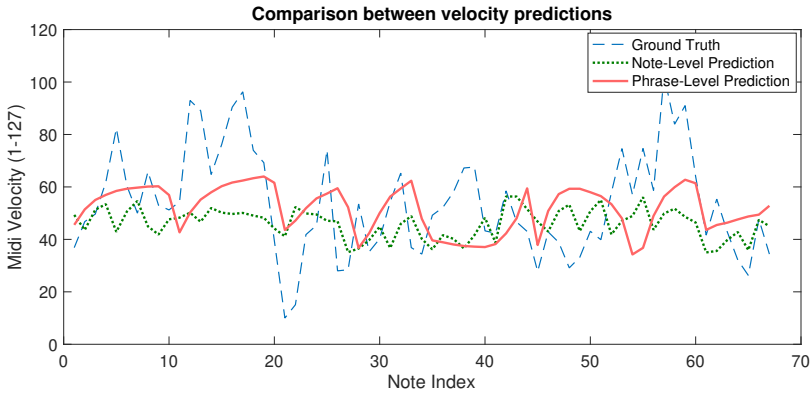


Figure 4.4: Velocity predictions across notes in a violin piece vs. performed ground truth.

The effects of a larger training set over the phrase-level modeling in its ANN formula-

Output coefficient	Pearson's r	
	(test set)	(training set)
x^2	0.2177	0.5222
x^1	0.2375	0.7054
x^0	0.2383	0.6818

Table 4.3: Correlation coefficients for output features.

tion are presented in tables 4.3 and 4.4. Table 4.3 shows correlation coefficients between predictions and ground truth values for all model outputs, each of which is a coefficient of the polynomial curves approximating phrase loudness in the performances. Values obtained for both training and test sets are informed for better discussion of the modeling limitations. Table 4.4 provides a more concrete measure of predictive accuracy, in terms of the mean RMS error in loudness values across all test set performances as well as the reference values of the same quantity for a deadpan performance and for the ground-truth values of output features, which correspond to the best-case scenario using parabolic curve approximations. Three prediction examples which we found representative can be seen in figure 4.5.

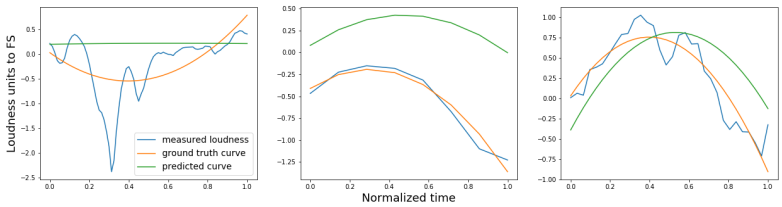


Figure 4.5: Comparison of loudness values measured in performance, their ideal (ground-truth) approximation, and model output for three phrases.

Lastly, in the perceptual analysis, a total of 305 pairs of melodies were compared by listeners in terms of human-likeness and personal preference. The mean perceived distinction between pairs was 3.41 ± 0.13 (on a 1-5 scale, $\alpha = 0.05$). Figure 4.6 shows the results divided into the three possible pairs according to expressive character: (C1) choice of human-based expression over "deadpan" baseline, (C2) human-based over modeled expression, and (C3) modeled expression over deadpan. The top row includes responses

Prediction type	Error level (dB)
Deadpan performance	3.86
Ground-truth approximation	1.69
Model prediction	3.39

Table 4.4: RMS error in loudness levels prediction.

Comparison	Question	p-value
C1	Human-likeness	0.7500
	Preference	0.8016
C2	Human-likeness	0.1440
	Preference	0.1440
C3	Human-likeness	0.7500
	Preference	0.8378

Table 4.5: Measured p-values for all perceptually evaluated comparisons.

from all participants. A sign-test with confidence-level of 95% controlled for 5% false discovery rate using the Benjamini-Hochberg method fails to reject the null hypothesis in all three comparisons (p-values listed in table 4.5), thus indicating that none of the versions was perceived as significantly more human-like nor preferred by users consistently, This is unexpected particularly for the comparisons of the first column which don't involve the model. Results shown in the bottom row help to provide some insight about the test setup by showing aggregate results of all comparison classes when considering exclusively the responses given by musically active participants of the survey, here characterized by the subset of people who reported practicing an instrument for at least one weekly hour.

4.4 Discussion

Comparing the results shown in each frame from figure 4.3, the variance of loudness signals is higher in the ESV dataset than in the EEP dataset. This reflects the varied charac-

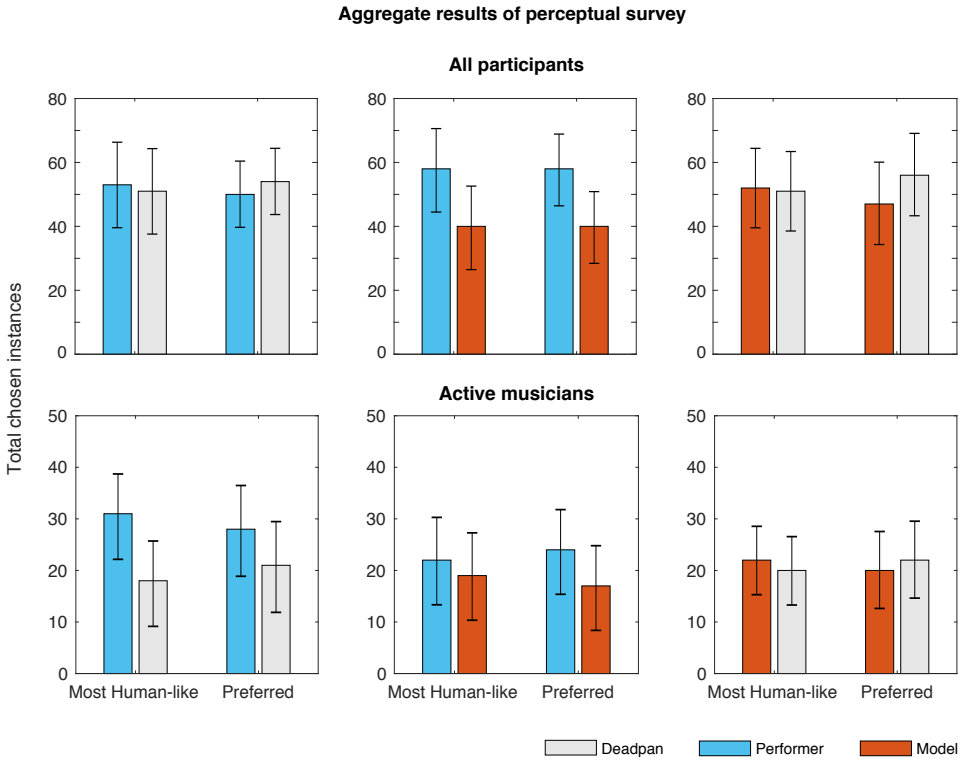


Figure 4.6: Results of perceptual survey pairwise comparisons.

ter of piece excerpts which comprise the ESV dataset, which translate to a poorer predictive ability of the phrase similarity model in this scenario. Still, the results show very similar error distributions between parameterized and interpolated (non-parameterized) implementations on both cases, which confirms that the characterization of phrase loudness in terms of mean value, dynamic range, and parabolic contour is a proper model design choice. Increasing the number of neighbors considered from one to three is effective at pruning out eccentric predictions as can be seen by the shorter tail of the distributions, but has been observed to also reduce the overall dynamic range of the output, making renditions a bit "dull". In fact, this effect is expected for small datasets such as the ones tested, since there ought to be very few examples of sufficiently similar melodies to be selected as nearest neighbors. Consequently, in such conditions, employing a single nearest neighbor is the most promising approach for a perceptually valid output, since the most melodically-similar phrase represents the available data point most likely to

have applicable expression data for a given target melody, and copying such data parameters from the same sample retains the coherence between different expressive output variables, an important trait when including timing information in the generated performance. For this reason, and considering the success of the parabolic representation as a parametric model of contour, the parameterized model with $k = 1$ was used for the synthesis of modeled performances for the perceptual evaluation.

Three of the four algorithms tested on note-level were able to outperform our phrase-level model in terms of mean absolute errors in performance prediction, and all except the note-level ANN exhibit some prediction success if compared to an inexpressive baseline. The poor rank of our model is a logical outcome, since the mere fact of operating on a note-level removes modeling restrictions; yet, the phrase-level model does not trail by a large margin and even overcomes the ANN, which is likely a consequence of the dataset size, that is much too small for satisfactory neural network training. These results are encouraging, since they indicate that the majority of the relevant information for predicting expression present in the note features was retained and summarized in the phrase-level form of the dataset.

Visually inspecting the velocity values predicted by the best-performing model, using random forests, against our phrase-level predictions and ground truth values from a recording of Edward Elgar's *Chanson de Matin* (fig. 4.4), it can be seen that the note-level model captures the oscillations in dynamics that happen between adjacent notes whereas the phrase-level model predicts smooth transitions. This partially explains the observed difference in performance, but works in favor of the phrase-level model in our valuation, since, in the intended pedagogical application, quick transitions are of little use as performance guidelines to students.

The predictive analysis over the larger MusicNet dataset highlight key strengths and limitations of phrase-level performance modeling. The gap between training and test set values for Pearson correlation coefficients seen in table 4.3 can be interpreted as overfitting, still, regularization attempts were not successful in increasing prediction accuracy indicating we've hit an accuracy ceiling for the given performance information under this representation. The deadpan-level (E_d) and ground-truth-level (E_g) errors in table 4.4 can be seen as lower and upper boundaries of accuracy, indicating that this modeling approach offers a potential reduction in prediction error of up to $E_d - E_g =$

2.17 dB. The 3.39 dB value obtained with our predictions implies an error reduction of $E_d - 3.39 = 0.47$ dB compared to the deadpan baseline, which corresponds to $0.47/2.17 = 21.65\%$ of the predicted potential. That is consistent with the correlation coefficient values and shows that the prediction of coefficients translates well into prediction of dynamics levels.

These results can be better interpreted if we compare them with the similar experiments of the non-linear, and best-performing, implementations of the models by Cancino-Chacón et al. (2017), also evaluated for their loudness prediction accuracy on a dataset of symphonic performances. The authors report the coefficient of determination (R^2) obtained for different design configurations, with values ranging from 0.205 to 0.282, which reflects a maximum explained variance of 28.2%. This makes the 21.65% error reduction obtained with our model all the more impressive, because their system is designed to parse and interpret the indications of dynamics present in musical scores, whereas our proposal generates performance features based on note information alone.

Another limitation of the MusicNet dataset used for our evaluation is worth highlighting: the provided symbolic music information provided (used in place of scores) makes no differentiation between first and second violins in ensembles where more than one voice for that instrument exists. This creates some distortions in our input feature set, since the actual melodies become mingled with a counterpoint voice, potentially interfering with the learning.

The prediction examples highlighted in figure 4.5 illustrate some relevant conclusions: It can be seen that most of the short-term variation in loudness levels occurs on note boundaries due to note articulation, and in terms of perceived dynamics can be understood as noise. The quadratic approximation (labeled ground-truth) provides a cleaner and more intuitive visualization of the variation of loudness in a phrase, and in most individually inspected cases represents it quite well. The leftmost example is an exception, as it shows a case in which the phrase boundaries chosen by the algorithm don't seem to match the performer's choice, hence the silence during the phrase and the poor results even in the proposed ground-truth approximation. In many observed cases, as shown in the middle and rightmost graphs, the predicted curve shows robustness, especially with relation to the x^2 and x^1 coefficients, since some variation in their predictions doesn't affect the character of the interpretation. It is reasonable to assume that despite

the difference between ground-truth and predicted values in such cases, performances executed according to instructions from the latter could be considered just as pleasing.

Regarding perceptual analysis results, typically (Katayose et al., 2012; Bresin and Friberg, 2013), one would expect a wide dominance of human-based renditions over inexpressive ones, which was not verified in the results of C1. The inclusion of such cases in the survey was intended as a mechanism for validating the experimental setup, since the corpus of existing results in this field has been based mostly on piano works, and, to our knowledge, no similar setup has been investigated for violin pieces.

Ratings for the measure of perceived distinction between audio clips was generally high across all comparisons. For C1, in particular, its mean value was 3.31 and standard error, 0.12 (on a scale of 1 – 5). This shows that participants were able to perceive differences in the renditions, but still reached conflicting decisions. Reflecting upon this fact and contrasting the melodies present in our dataset against pieces typically found in benchmark datasets (e.g.: as used by Oore et al. (2018)) point to two main causes for participant disagreements. First, the lack of some expressive features such as articulations, vibrato, and timbre variations, which often work in conjunction with the modeled features, facilitating their interpretation. In particular, since loudness values for all synthesized pieces were determined on a note-level, variations in attack-time were ignored. This can lead to discrepancies between nominal note onset time and the perception of when a note actually begins, changing the interpretation of timing variations. A second likely cause for participant disagreement in C1 is the use of pieces written for an ensemble (namely violin and piano) without the accompanying instruments, which removes the melodies from their contexts making it harder for listeners to parse their structure.

This view is reinforced by some participant comments. One of them states, after declaring preference for the deadpan rendering over the human-based one: *"Little big ambiguous; A is more flat and regular, but it kind of depends on the context whether this would be appropriate or not."* A, in that case, being the deadpan performance. Another one commented: *"I prefer the dynamics in B and the time in A. It's easy to distinguish them, but no one sound more human than the other."(sic).* In this case, B was a human-based rendering, and A, a deadpan one. Some comments can also be found which favor the modeled rendering, e.g.: *"I prefer the dynamics in A and the time fluctuation in B."* for a comparison where A corresponds to the modeled rendering, and B, to the human-based

rendering.

From the musicians' results graph, it can be seen that the percentage of choices favoring the deadpan renditions is smaller in this subset than in the full result set, which could reflect a higher ability among this group in interpreting the performances in the context they were presented. Furthermore, the percentage of choices favoring the modeled performance is larger for the musicians' data than for the full set, which can suggest that, moving past the limitations of the listening experience, the modeled expressive variations are aligned with real musicians' expectations.

Despite these challenging conditions, the experiment confirms that using only around 6 minutes of total reference audio time, our model was able to introduce high ranges of expressive variations in the synthesized pieces which were consistently perceived as different from other renderings but not consistently rejected. The absence of a clear tendency towards the human-based synthetic performance, however, prevents us from stating stronger claims.

4.5 Conclusions

The phrase-similarity model has produced some convincing suggestions of expression, at times worthy of praise by listeners in a blind setting, with considerably less training data than most state-of-the-art models and virtually no time expenditure on model training thanks to the musically coherent approach of processing phrases rather than isolated notes. For the desired pedagogical applications, the ability to produce musically valid expressive performances from few examples gives it versatility, allowing students and teachers to select the most relevant reference recordings to make up a training set, for instance for studying the style of a particular performer, or of a specific musical genre. Additionally, the smoothness inherent in the curves output by our model makes the expressive movements represented by them much easier to follow in real-time by a student or performer.

We have been able to explore a little of the model's creative potential during the 10th International Workshop on Machine Learning and Music (MML2017), when we showcased a live performance based on generated dynamics suggestions. In this event, a traditional Puerto Rican song was performed on the violin as the musician followed vi-

sual directions of dynamics generated by our phrase similarity model trained on a small corpus of performances of the double violin concerto by Johann Sebastian Bach. The software used to provide real-time feedback for the performer was an early version of SkyNote, which will be the object of study in chapter 6.

The proposed automatic phrase segmentation algorithm was successful enough to promote a model that explains roughly 21% of the dynamic variance in pieces when given a wider variety of samples, besides improving the sense of structure in generated performances. However, its static nature also limits the improvements that can be made to our model, negatively impacting the learning in cases exemplified by the leftmost phrase in figure 4.5. As we've seen in the evolution of piano-based CSEMP as well as automatic composition systems discussed in chapter 2, as more and better-quality data and algorithms become available, it also becomes more fruitful to incorporate the learning of input features to the training data processing rather than imposing a certain structure that destroys some information as it organizes the remainder.

Lastly, when contrasting the performance of phrase-level against note-level modeling, our phrase-level approach was able to achieve comparable results despite the resulting information compression that comes from summarizing note features in terms of phrase similarity.

Chapter 5

Performance Modeling by Deep Learning on Note Sequences

5.1 Introduction

In the previously presented models of performance, in order to ensure an emphasis on the expressive variations that stem from musicians' deliberate and long-term execution plans – our primary interest, as presented in chapter 1 and highlighted via the conductor analogy – our model designs actively manipulate the information presented to the core learning algorithms, filtering the inputs and placing restrictions in the outputs justified by musicological assumptions. The most notable of those manipulations is the summarization of musical content into phrases, inducing the algorithms to learn patterns tied to the musical structure of pieces, and which were therefore retained in the information related to phrases.

The unfortunate consequence of these design strategies is that their methods of data summarization are imposed rather than learned from the performance datasets. This, combined with the static design of the algorithms with respect to the specific musical elements of each piece meant that the proposed preprocessing of musical pieces for input and output feature generation doesn't always properly represent the music that originates it.

The best concrete example is the phrase segmentation algorithm itself, as presented on

page 40. Even though our automatic solution using the LBDM is able to produce a musically acceptable partition of the score, when paired with data from a specific performance, such a representation may fail to capture the phrasing intentions of the performer, since, as famous violin teacher Leopold Auer put it, "no two artists phrase the same passage in exactly the same manner" (Leopold Auer, 1920). These limitations can also be observed with respect to the harmonic features of phrases as compiled for our ANN model, listed on table 4.1. For instance, the estimation of chords, keys, modes, and dissonance are all computed from the contrast between a given phrase and the entire piece, which loses its meaning in the presence of key modulation, modal mixture, or in pieces from different periods or traditions that do not adhere to classical tonal harmony.

These data inconsistencies are worsened by the inability of the previously presented learning algorithms to take the context of data samples into account. The patterns learned by k-nearest neighbors or multi-layered perceptron models are always a function of the information present in each individual data sample, in our case, a single phrase. The hierarchical nature of musical phrasing, however, demands a broader view of piece sections. The class of learning algorithms best suited for this comprises the *Sequence Models*, developed primarily for properly parsing language in texts, and discussed in sections 2.1 and 2.1 along with their applications to music.

This chapter presents an effort to develop a new model of performance which can address the liabilities discussed above by means of deep-learning architectures which incorporate the context of input samples into the learned optimization functions. As we mentioned in our review of deep-learning designs, these systems are marked by an ability to simultaneously learn the target function and the most adequate input representation for the task. Given this theoretical skill, we take the opportunity to investigate whether models of musical performance built on score information using such techniques are capable of learning optimal phrasing from the dataset itself on a note-level, without resorting to previous processing as we had done, thus avoiding the pitfalls associated with those fixed decisions.

With this broad objective in mind, the experiments described next pursue following goals:

- To assess whether the recent advances in language and sequence models are suited as a design strategy for expressive performance modeling;
- To define a system for encoding musical information that provides optimal results in the machine-learning task of our problem domain;
- To study the sensitivity of the produced outputs with respect to model and dataset sizes;
- To offer insight into the specific challenges of modeling violin performance and the value of the proposed output features as descriptors of performance expression;
- To allow a comparison between the characteristics of expressive suggestions obtained from note-based sequence models, and the previously proposed phrase-based models.

5.2 Method

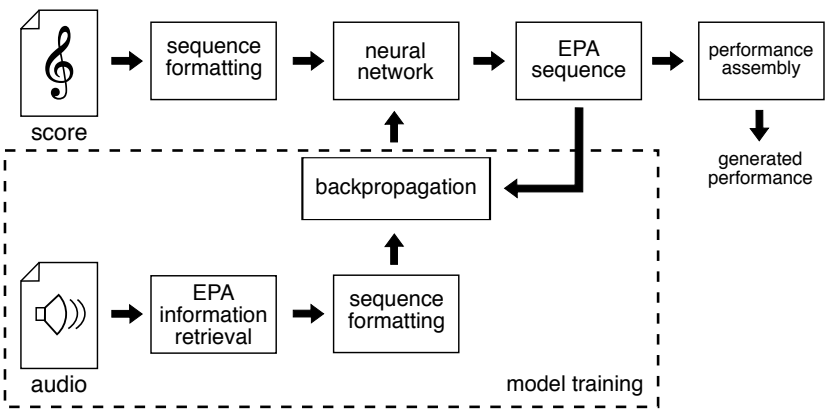


Figure 5.1: Overview of the processing steps involved in the note-level sequence model of performance.

The structure of our sequential model of performance is outlined in figure 5.1. The information in scores is translated into sequences of symbols from a machine-readable

vocabulary by the *sequence formatting* module. These sequences are processed by a carefully designed recurrent neural network which generates sequences of note-level data on the chosen expressive performance actions. During training, these sequences are contrasted against performance data extracted from reference audio recordings of the same musical pieces, computing the loss function which feeds the neural network's backpropagation algorithm. During operation, the outputs of the neural network can simply be compiled as a new performance, be it for a synthesized rendition or analytical purposes.

Next, we provide further details about the most sensitive design choices in these modules.

Data encoding

Even though, owing to their representation learning capabilities, deep learning models require less preprocessing of input data compared to traditional machine learning algorithms, the encoding of musical scores into a sequence of pre-defined symbols is both necessary and significant to the system's performance. Typically used languages for representing musical compositions such as the MIDI file format or MusicXML are not yet suited to these tasks. This is due in part to their complex grammar, but also their verbosity – neural network designs for language processing such as RNNs see best results with data sequences spanning roughly under 100 symbols. As a consequence, systems centered around learning musical languages adopt representations with a higher information density. The encoding structure adopted by PerformanceRNN (Simon and Oore, 2017) and shared by the Music Transformer (Huang et al., 2018) is an apt candidate for its similarity to MIDI, concision, and good results. This encoding includes different symbols for indicating note onsets or offsets in each specific pitch, as well as separate symbols which indicate the passage of a certain amount of time in milliseconds, and a third set of symbols denoting note dynamics within a discrete range. This creates a finite vocabulary that fully represents the musical performances, which the neural network learns to convert into a real-valued vector encoding the necessary information for its task.

Our model uses an adaptation of the performanceRNN encoding that takes into consideration that rhythmic and melodic note information are naturally independent, and by virtue of that it is possible to facilitate the representation learning task of the neu-

ral network. Instead of representing the passage of time by defining different symbols for quantized time advances, and representing note durations by separating onset and offset events, we instead limit the language vocabulary to a single type of event – note onset – having different symbols for each pitch within the well-tempered scale present in the dataset. We then include the numerical features of those notes, namely the number of beats elapsed since the last event and the number of beats until that note's offset, as extra dimensions concatenated with the *embedding* vector, that is, the real-valued vector which is the learned representation of that language symbol. Figure 5.2 illustrates the process in question comparing it to a "typical encoding" in the style of PerformanceRNN. Note velocity is not included because our encoding is not meant to represent performances, but scores without any expressive guidance. The extra feed-forward layers in our design are aimed at unifying the representations with information shared between both the learned pitch embedding and numeric rhythmic features.

Our encoding scheme has a few theoretical benefits over the PerformanceRNN design. Total sequence length for representing a musical piece is reduced. The one-to-one correspondence between input vectors and notes also simplifies obtaining some crucial information about notes, such as its duration, which in performanceRNN is spread across many inputs. Additionally, our process eliminates the need to discretize the timing information, and the necessity of parsing grammar is also virtually eliminated, as any sequence with values within the proper domain is a well-formed musical piece. As a consequence of the final reduction in vocabulary size, the number of network parameters in this section is kept lower, which facilitates training and reduces memory requirements.

This decision of inducing the learning of separate representations for independent input features also allows us to evaluate the impact of introducing some complex harmony-related information in the model, as we will detail when discussing the model evaluation. Finally, we should note that in the datasets that required so, this representation was supplemented by an identification of which musical instrument produces each note in question, encoded in the form of a one-hot vector of possibilities and concatenated with the rhythmic features.

Unlike many natural language processing models, the inputs and outputs of our system do not share a format. Our model generates note-level information on performance dynamics – either represented by MIDI velocity or loudness values, according to the

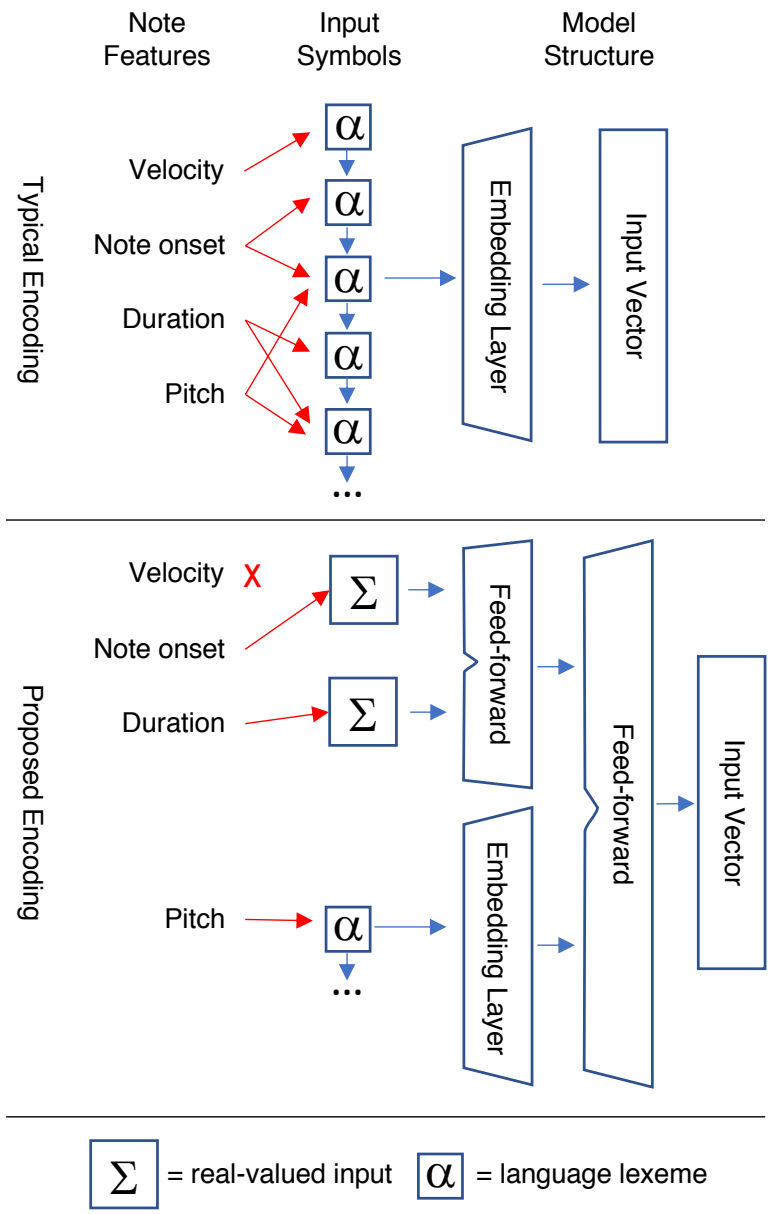


Figure 5.2: Structure of the model input encoding, compared to a typical design.

dataset – and also articulations as variations of note durations and, lastly, performance timing – represented by inter-onset interval ratio (IOIR), computed for each note n as follows:

$$IOIR_n = \frac{onset\_seconds(n) - onset\_seconds(n-1)}{onset\_beats(n) - onset\_beats(n-1)} \tag{5.1}$$

All numeric features of the dataset are normalized so that they have zero mean and unity variance within each piece, including the output features of the training set. This means that, similar to the models presented in chapter 4, some features of the desired piece performance should be defined prior to using the model, the most significant of them being mean loudness, dynamic range, and tempo.

Sequence preparation

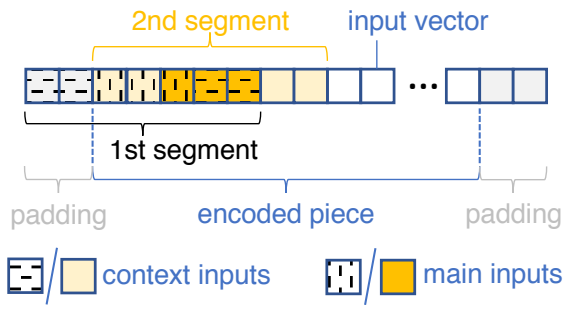


Figure 5.3: Diagram of the sliding window mechanism of input sequence partitioning.

Despite the sequence length compression achieved by our data encoding scheme, the number of notes in each complete score from our corpi still far exceeds the maximum amount manageable by sequence models within the memory and training time constraints of present-day computer architectures. The customary solution in natural language processing models is to set a fixed sequence size and then process the input data in batches of that size. In order to differentiate the real start and end of a sentence from the starts and ends of batched segments, special *start-of-sentence* and *end-of-sentence* symbols are typically included in the encoding language.

Given the importance of smoothness and continuity in expressive performance features, that approach raises concerns about the quality of generated outputs near the boundaries of batched segments, since the model is given no knowledge beyond those boundaries – a problem already observed in our phrase-level modeling approach.

Our proposal to prevent this issue is to replace the piece partition with a *sliding window* of input sequences, so that each segment includes extra inputs before and after the target sequence for which the model should produce outputs. This slightly reduces the efficiency of the computation, since some inputs are seen more than once by the model, but effectively gives it some context to guide the feature generation. In practice, the implementation of this mechanism is achieved simply by discarding the outputs produced by the model in the extremities of each sequence, both during training, excluding them from the loss function calculation, and during runtime. The division of each piece into smaller sequences must also account for the sliding window, including the proper overlaps and padding according to the context size configurations, as seen in figure 5.3.

Neural network design

The structure of the neural network powering our performance model is shown in figure 5.4. The input layers follow the design explained for our input encoding. A bidirectional gated recurrent unit (GRU) processes the whole input sequence, generating an encoded version of it. The output of this *encoder* portion of the network is then used in a *decoder* section. The decoder generates an output vector per sequence step, combining four information sources in a feed-forward network: the outputs generated in the previous step, the encoded input from the current step, the encoded score, and a state vector. The encoded score is not entirely sent to the feed-forward network, but summarized in a *context* vector via the self-attention mechanism, as implemented for the transformer model (Vaswani et al., 2017). The state vector is produced by another GRU, and takes into account the previous outputs as well as the context vector.

The training process was carried out using 90% of the available data in each dataset, of which 10% was also reserved as a validation set for fine-tuning network hyperparameters. Regularization was achieved interposing layer normalization and dropout layers in all network sections. Backpropagation was optimized with the Adam algorithm (Kingma and Ba, 2014). A *teacher forcing* strategy was also implemented for training the decoder:

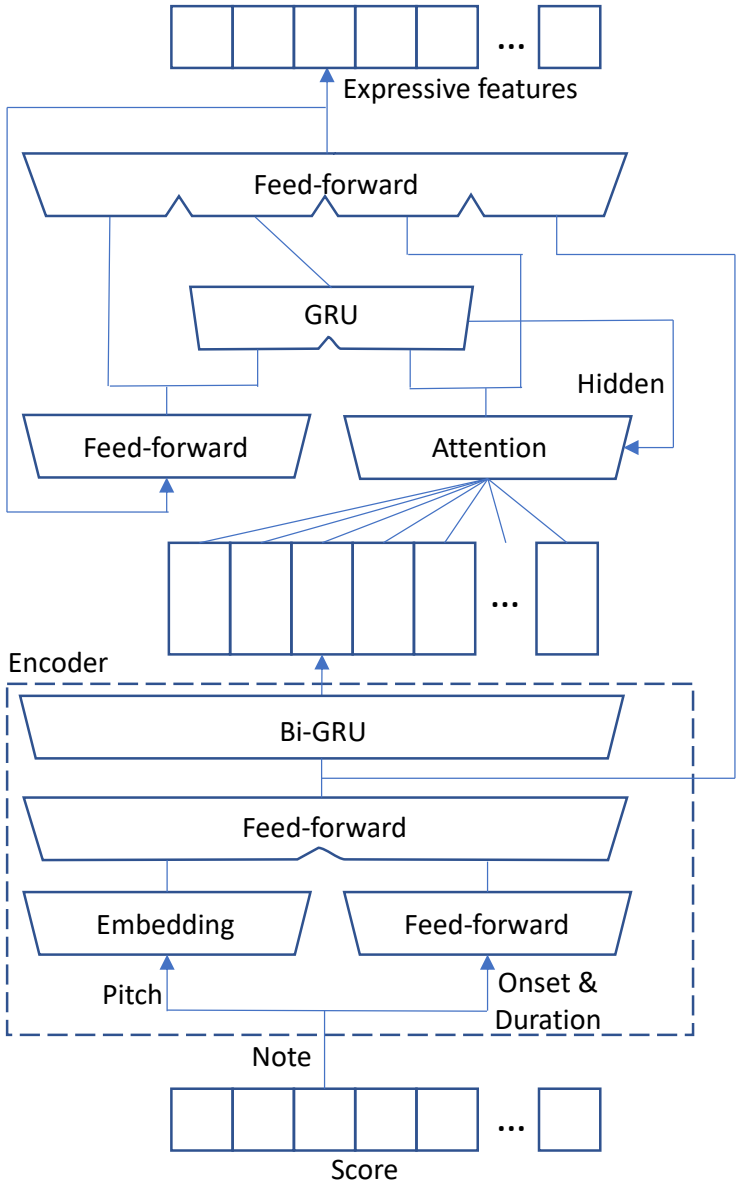


Figure 5.4: Architecture of the sequence model of performance.

for a fraction of the decoder training steps – 50% in our implementation – the previously generated output is replaced in the network by its ground-truth values, as if the

model had produced an optimal prediction. This process improves network weight optimization in recurrent networks.

Evaluation datasets

Several experiments were conducted with the purpose of validating the design decisions and assessing the quality of performances generated with our sequence model. The broad range of research questions we hoped to address and the specificities of each available dataset meant that a combination of them was necessary to cover all experiments.

The MusicNet dataset (Thickstun et al., 2017), which was previously used for evaluating of our phrase-level CSEMP as described in chapter 4, remains a good source also for training the proposed sequence models on performance timing, given that it contains not only the onset time and duration from notes of all instruments aligned with each recording, but also their original, score-mandated, values, measured in beats. Two subsets of MusicNet were prepared for model training. One of them consists of all pieces which contain a violin in the ensemble – the same set employed in chapter 4. The other one is composed only of all recordings of violin sonatas by Ludwig van Beethoven that were available, separated by movements. Since MusicNet doesn't include the full scores of its pieces, we have augmented this second dataset by including important extra information about the compositions easily found in the scores but absent in MusicNet: time signatures and measure separations.

Lastly, we also employ the MAESTRO dataset (Hawthorne et al., 2019), version 1.0.0 [1] for its large size and reliable dynamics information via velocity values. Table 5.1 highlights the differences between these datasets.

As a resource for artificially increasing the number of training samples, all datasets were augmented by including transpositions of all pieces up to 3 semitones below and above their original pitches, a technique borrowed from the Music Transformer model (Huang et al., 2018).

The code and instructions for building models, augmenting and processing the datasets, and reproducing the results below can be found, at the time of publication, in the au-

[1] available at https://magenta.tensorflow.org/datasets/maestro#v100

Dataset	Ensemble	Recs.	Notes	Features
MusicNet Violin (V)	various (chamber)	123	3.6M	I L ND NO P PD PO
Beethoven Sonatas (B)	piano + violin	21	600k	I L M ND NO P PD PO
MAESTRO (M)	piano	1184	6.2M	P PD PO V

I: instrument, L: ensemble loudness, M: measures, ND: note duration (score), NO: note onset (score), P: pitch, PD: note duration (performance), PO: note onset (performance), V: velocity.

Table 5.1: Details of the datasets used for evaluation.

thor's Github page [2].

5.3 Results

The first step in the analysis concerns itself with the influence of input sequence length in the overall accuracy of models. Figure 5.5 shows the coefficients of determination (R^2) obtained in a velocity prediction test using the validation set from dataset V for various configurations of sequence length. The best results were obtained for lengths ranging between 16 and 32 inputs, meaning, with our representation, 16 to 32 musical notes. In all tests, the network was configured with a base hidden vector size of $h = 256$ bits, 2 layers per GRU section, and dropout rate of 4%.

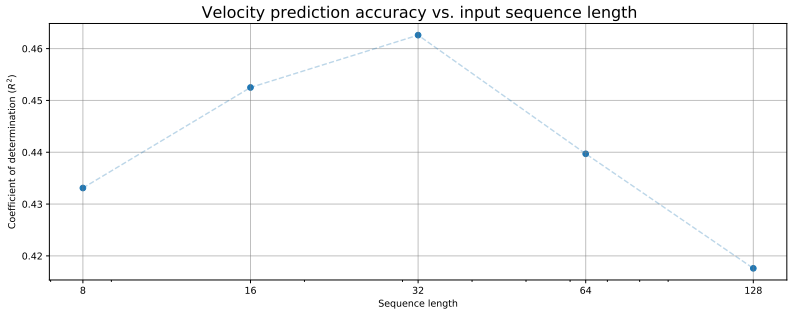


Figure 5.5: Influence of sequence length in model accuracy for dataset M.

Next, we move on to test the efficacy of our sliding window design. Table 5.2 provides the summary of velocity prediction accuracy tests using the same configurations as

[2] https://github.com/fabiozeh/deep-expression

above. The results confirm a slight improvement over the baseline configurations with no note overlaps (i.e: equal output size and sequence length values). The same effect occurs for timing predictions using dataset V, as indicated in table 5.3. In this case, best results occur for slightly shorter sequences.

Sequence Length	Output (Hop) Size	R^2 (%)
32	32	46.26
16	16	45.25
48	32	45.14
32	16	46.44
32	30	46.47
32	1	46.50

Table 5.2: Velocity prediction results from dataset M with various windowing configurations.

Sequence Length	Output (Hop) Size	R^2 (%)		
		Timing	Articulation	Loudness
128	128	77.75	61.07	-16.32
32	16	79.02	60.60	4.36
16	16	77.94	55.50	6.67
16	8	78.75	**62.84**	2.32
8	8	78.58	59.58	**9.95**
4	1	**79.21**	54.65	7.80

Table 5.3: Prediction results from dataset V with various windowing configurations.

Using dataset B we were able to analyze the influence of having a semantically complex input feature in model accuracy. Table 5.4 compares the coefficients of determination for all predicted EPAs using simple note descriptions and including a *metric strength* feature computed according to each note's position within the measure. Though prediction accuracy for inter-onset intervals saw improvement with the inclusion of metric strength information, the same was not true for duration predictions. Moreover, the negative effect of having fewer training examples in dataset B than in dataset V far outweighs the benefits of this feature.

Input Feature Set	R^2 (%)		
	Timing	Articulation	Loudness
Basic Note Description	51.30	66.65	-20.78
Above + Metric Strength	54.62	59.88	-18.97

Table 5.4: Prediction results under different input feature sets.

Finally, we focus on achieving optimal model configuration to put it into perspective against similar research efforts. Figure 5.6 reports prediction accuracy values with various network size configurations on dataset M, used for piano key velocity predictions. The horizontal axis corresponds to number of network parameters, and shows the expected trend of better performance for larger networks, up to a saturation point. All tests were run with a sequence length of 32 notes, and a sliding window with a hop size of 16 notes, resulting in 8 context notes at the start and end of each sequence. The same exploration was reproduced on dataset V, and a choice of configuration was made that provided the best combined results of all three output features. The accuracies obtained by the chosen configurations on their corresponding test sets are compared to other relevant models in table 5.5.

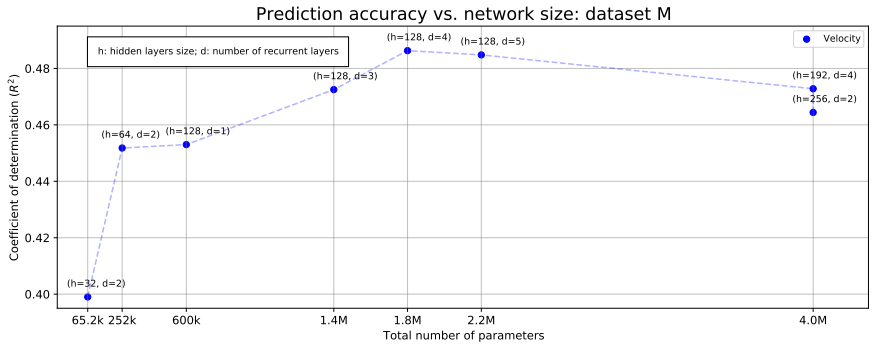


Figure 5.6: Influence of network size in model accuracy for dataset M.

Model	Ensemble	Notes	Samples*	Feature	R^2 (%)
RNBM†	orchestra	?	53.8k	Loudness	**28.2**
VirtuosoNet‡	piano	3.5M	3.5M	Velocity	46.8
				Timing	25.3
				Articulation	31.2
Ours (phrase-level)	various	3.6M	15.6k	Loudness	21.6
Ours (sequence)	piano	6.2M	6.2M	Velocity	**47.2**
Ours (sequence)	various	3.6M	3.6M	Loudness	3.4
				Timing	**52.7**
				Articulation	**59.9**

* Number of data samples after preprocessing.

† (Cancino-Chacón et al., 2017)

† (Jeong et al., 2019a)

Table 5.5: Expressive feature prediction results across several models.

5.4 Discussion

The analysis of generated EPA features as performance predictions reveals that the structures learned by the neural network do generalize to pieces outside the training set. The measured accuracy in predictions either approaches or surpasses state-of-the-art models across all features except ensemble loudness, and it does so without needing information on expressive score markings – a unique trait among recent approaches.

On the low R^2 from loudness predictions, it should be noted that, unlike the RCO/-Symphonic dataset used for evaluating the RNBM model, listed in table 5.5, the performances in datasets V and B were recorded in various circumstances, thus having noticeable differences in audio quality. Although predictions were attempted for normalized features, which attenuate such variations, differences in compression levels resulting from specific recording conditions could decouple measured loudness from performance dynamics even further, leading to inconsistent training examples that would prevent proper optimization. Nevertheless, the higher accuracy obtained by our phrase-level model in the same feature of this dataset supports the design choice of summarizing ensemble scores to facilitate network training – a strategy that was also adopted in the

RNBM model.

The influence of sequence length in prediction accuracy, seen in figure 5.5, reveals improvement as sequences grow and notes are processed in context, but model quality quickly degrades beyond 32 notes per sequence. This degradation is expected beyond a certain point in all applications of sequence models, even in modern implementations, due to the vanishing gradients problem (Hochreiter, 1998). A length of 32 symbols, however, is still a modest number for GRUs, and the loss of accuracy beyond this point reinforces our belief that the expressive content of performances is best observed on a *phrase-level*, typically spanning one to four bars, and consistent with the best-performing lengths observed in training. In fact, a length of 32 symbols is enough to encircle a typical phrase spanning one to two bars and its surrounding notes, informing the model about a few elements of the context in each phrase. Combining this with our sliding window system is a very logical decision to mitigate the issues faced in our phrase-level model caused by arbitrary phrase separations.

Sequence models in their current form are still unable to process an entire piece at this granularity level. This also presents itself as an issue in text generation tasks, where model outputs struggle to maintain coherence over several paragraphs. VirtuosoNet tries to work around this limitation learning intermediate representations for a hierarchical architecture, but cannot achieve much lower prediction error levels. There is evidence that this hierarchical treatment improves performances perceptually, though while the improved sensation of quality is not properly quantified, it is difficult to incorporate their benefits to the training algorithms for taking them further.

The inclusion of complex input features attempted with the training of dataset B demonstrates that the preference for high quantity, raw data over a smaller, preprocessed dataset when training deep-learning models also proves true in the domain of music performance.

Similarly, the exploration of various network sizes confirms that our problem domain, just like deep-learning models in general (Bengio et al., 2021), produces more accurate models with increasing network sizes, particularly in depth. The results show, however, that a limit is reached according to our dataset size, beyond which the model's ability to generalize stops increasing. Unlike datasets of other fields, such as CIFAR-

10 (Krizhevsky, 2009), in which one knows that the training set samples contain enough information to properly classify the entire test set, it is impossible to map all the influences and thought processes that lead a musician to perform in a given way. Moreover, our training examples represent only a small fraction of the musical repertoire, even if restricted to the western classical canon.

Taking these models forward likely goes through the adoption of techniques from statistical modeling, as most similar works have proposed. This type of treatment can elegantly incorporate the many possible variations in expression stemming from musicians' creativity, which place an upper bound on the accuracy of deterministic mappings as we have pursued. We still believe that applying deterministic methods is an opportunity to test where this upper bound lies, and indeed we have observed that high accuracy in EPA prediction can be achieved even without information from score markings regarding expression, such as dynamics and articulation annotations. Ignoring this type of score markings was a deliberate decision, because we were interested in producing models which can induce expressive variation in the absence of directions, which is often necessary for musicians learning from lead sheets, for instance. Therefore we didn't want the absence of expressive markings to be confused for an indication to play in a certain (monotonous) way. Still, using datasets that can differentiate when expressive markings are available or not is a known possibility for improvement.

Chapter 6

Technology-Enhanced Expressive Performance Practice

6.1 Introduction

This chapter describes the pilot evaluation of a technology-enhanced setting for the practice of expressive performance, implemented on top of the SkyNote software for violin learning, which was developed as part of the European project TELMI (Ramirez et al., 2018).

In the proposed setting, the practicing performer is equipped with a computer interfaced with an ambient microphone. Using the software, they can view the music score in MusicXML and rehearse in either of two modes: free or guided performance. In both cases, the software analyses all recorded audio from the microphone, providing visual feedback about the performance. In free performance mode, this feedback is provided post-hoc; there is no interference from the software during play. Guided performance mode, however, is designed to allow the musician to play along with a reference recording while receiving real-time visual feedback about the expressive quality of both the reference, and their own interpretation.

The concept of this scenario is an attempt to use technology to find a middle ground

between pedagogical techniques typically employed in the teaching of musical expression, previously discussed in chapters 1 and 2. Guided practice as proposed has potential to provide the precision of auditory modeling with the clarity of technical instructions. The combined and consistent application of a recorded reference supported by feature-related visual aid addresses some of the reported shortcomings of those teaching methods, since the visual cues indicate what the student should listen for, and the impact of specific playing techniques can be heard, seen, and contrasted against a target.

As for the origin of reference performances for guided practice, there can be many sources. The most logical and practical one can be recording one's music tutor, if the option exists. Commercial recordings are a viable alternative, provided that the parts to perform are prominent and clearly audible. A third option worth highlighting is the application of a CSEMP, crafted in line with the conclusions previously stated in this thesis. If we extend the abstraction, we can apply the CSEMP model by Kirke and Miranda (2009) to the whole scenario and consider the human-computer system. As the musician practices to the generated reference, they can adapt their own performance by reflecting upon the contrast between their expressive choices and the generated ones, effectively assuming the role of the adaptation process module. More than teaching students to play in a specified way, this process can be the catalyst of a broadening in their musicality.

The goals of the evaluation described next are one step behind the use of modelled performances, though. We are primarily interested in exploring the responses of musicians with various profiles and understanding how each can benefit from a system such as this, particularly with regard to the strengths and weaknesses of the proposed learning scenario in terms of the proposed software features, their technical implementation, and the impact of this pedagogical approach. To that end we have conducted a small-scale pilot study where violinists of varied proficiency levels and backgrounds are asked to practice imitating a performance to the best of their ability, either having only score and audio as tools, or having the support of our software. Later, these participants are queried about their perception of the task, and their impressions on the software. We believe that this method is ideal for collecting qualitative feedback and creative suggestions because the proposal is concrete and testable, even though its implementation is in an early stage.

In a nutshell, the goals of this experiment were the following:

- To propose and prototype a technology-enhanced platform and method for music expression learning and practice consisting of visually and aurally assisted imitation exercises;
- To investigate how musicians of various profiles respond to this scenario, and how it may benefit each one of them;
- To evaluate whether the technical solutions for audio-to-score alignment, intonation and dynamics recognition, and dynamic tempo adjustments were properly chosen and implemented in the software and well-received by users;
- To collect user responses and views on the performance feedback tools and visualizations provided for the exercises in the software prototype;
- To gather evidence for a wider debate about the utility and effectiveness of technology-enhanced methods for expressive performance teaching.

6.2 Tools development

We detail, for the sake of clarity and reproducibility, the features that were added to the SkyNote software, and how they were programmed. In its original concept, SkyNote was intended as a violin practice tool that could provide automatic feedback on technical aspects of performance using audio and motion capture devices. As such, the exercises designed for it were essentially about violin technique, intended to be played with metronome, and not particularly musical. Figure 6.1 shows the score for one of these exercises, which was accompanied by instructions which read:

> In this exercise you will vary your *contact point* while keeping your bow weight, tilt and speed constant.
>
> Notes in *standard* notation are played *near the bridge* and notes notated with an *"X"* are played *near the fingerboard*.
>
> Each bar is played with *one bow*. Play this exercise on the *A string* with your *3rd finger*.

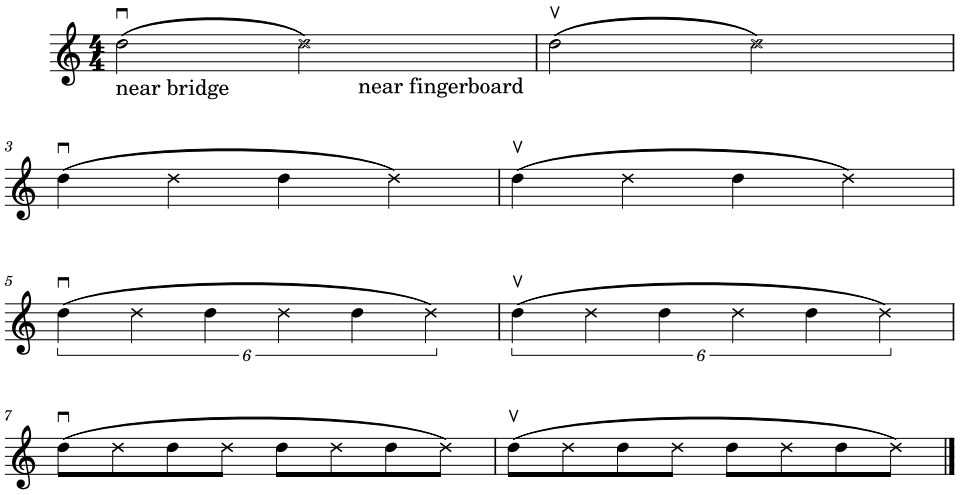


Figure 6.1: Example of exercise from SkyNote's original repertoire.

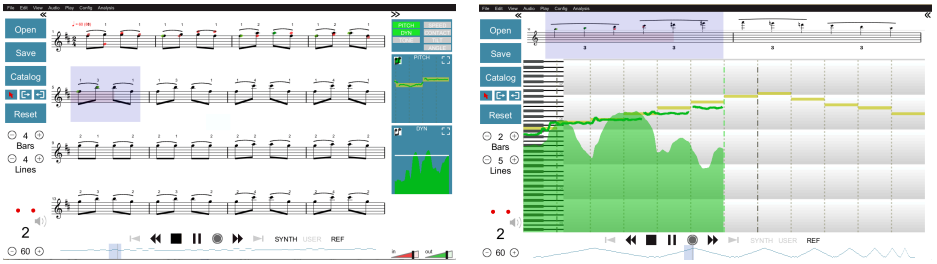


Figure 6.2: Original score and piano roll visualizations from SkyNote.

Naturally, monitoring the practice of expression in performance has very different technical requirements. Still, we identified the possibility of developing a platform for that goal by repurposing the existing score and piano roll visualizations, shown in figure 6.2 and relying on the pitch and volume recognitions obtained via audio processing. To complete the technical requirements for the two desired modes of practice – free and guided performance – it was necessary to solve two technical issues related to abandoning the use of a metronome: *audio-to-score alignment* of a free performance, and mid-performance tempo changes, or *"dynamic tempo"*, for playing along a reference.

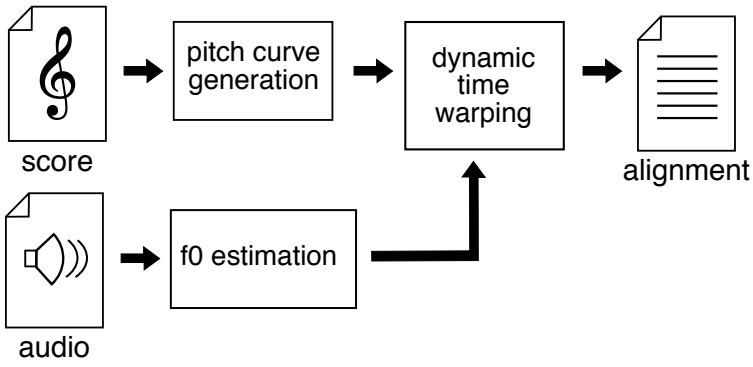


Figure 6.3: Structure of the audio-to-score alignment algorithm.

Audio-to-score alignment

Our analysis of music performance must begin with the association between segments of the recording and the score. Since the original SkyNote exercises required following a metronome, that association was purely arithmetic, as each note was expected to be found at specific recording times. In an expressive recording, the tempo and timing variations require a more sophisticated logical mapping. Even though real-time score following systems can be developed with good accuracy (Cont, 2008; Nakamura et al., 2013), the planned use cases for this study allow us to opt for a simpler solution.

The implemented audio-to-score alignment algorithm is represented by the block diagram of figure 6.3. Every recording in *free performance* mode is subjected to a *post-hoc* analysis by this algorithm, resulting in mapped onset and offset times for every performed note, and allowing the system to provide the same feedback on note correction and intonation as with the metronome-based exercises, visible in figure 6.2.

The algorithm works by comparing an ideal pitch curve for a steady-tempo performance with the fundamental frequency ($f0$) estimate extracted from the performance using the analysis algorithm proposed by Serra and Smith (1990). Using dynamic time warping, the optimal alignment between the two curves is found, indicating the best correspondence between score and performance. Robustness has been observed to improve

with the calibration of a few parameters: pitches are considered equal if their frequency lie within 50 cents of one another; pitches which differ by perfect octaves are also considered matching; warping penalties are set at the same value for skipping samples – indicative of inconsistent tempo – and for pitch mismatches during rests, and four times as much for other pitch mismatches – indicative of mistakes, ornamentation, or incorrect pitch estimation. Figure 6.4 shows an example of alignment obtained with this method. Each grey lines shows the resulting mapped onset for one score note.

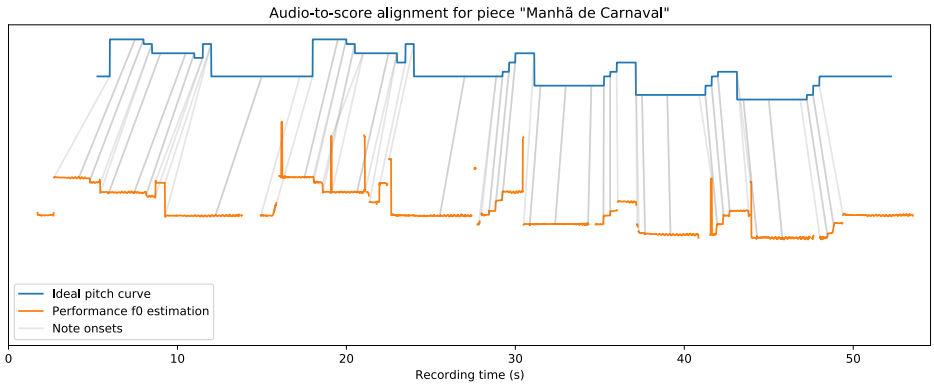


Figure 6.4: Audio-to-score alignment example.

Dynamic tempo changes

For real-time accompaniment of a reference performance, as desired for a guided performance practice, besides knowing the correct note onset and offset instants in a recording, the tempo at every instant must also be known. This affects many things, from the correct overlaying of reference and real-time graphs to the length of notes drawn in piano roll view. Our solution to this issue assumes that changes in tempo in a performance occur as smoothly as possible while still ensuring that the onset times detected with DTW are correct. This is accomplished by constructing a transformation from ideal, steady-tempo time and actual performance time as a monotonic piecewise cubic spline interpolation, using the method by Fritsch and Carlson (1980). The graph in figure 6.5 shows an example of this function for a section of a performance of the piece "Manhã de Carnaval" in which some tempo variation occurs. The Y-axis shows time in seconds as expected if the song were played with metronome, whilst the X-axis shows real record-

ing time. Deviations from the diagonal are caused by expressive variations. Orange dots represent the moments when note onsets happen, which were used as knots for the interpolation algorithm. The graph shows that the interpolation is indeed smooth and monotonic – only moving forward in time – and passes through all designated points. This transformation also allows us to introduce a "dynamic metronome" for guided performance practice – this feature can show a pulsating light and play click sounds on beats synchronously with a reference performance. This is equivalent to having a metronome which varies its tempo in real-time to follow a musician.

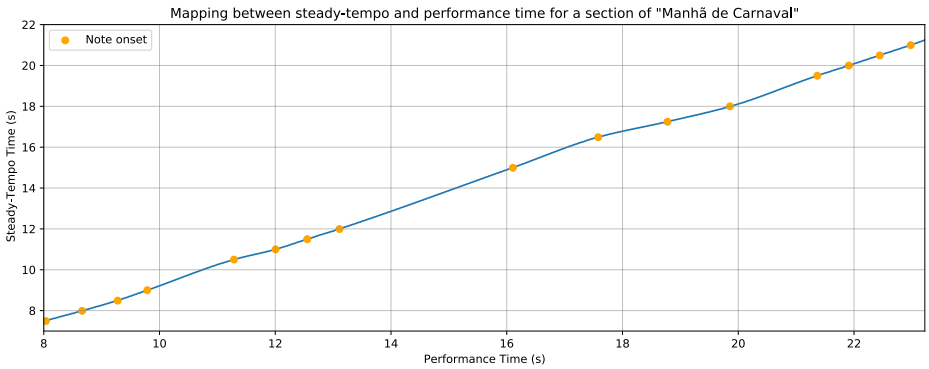


Figure 6.5: Example of dynamic tempo mapping function.

Besides these main additions, extra attention was given to the calibration and display of input microphone loudness values for its importance as visual feedback on dynamics. We have moved from a measurement in RMS signal energy to the LKFS scale introduced by the EBU R 128 standard for its close relationship to loudness perception, and we have improved its visibility when shown next to the score.

With the solution of these technical necessities, the resources available for performance practice in the software are:

- Performance report indicating wrong intonations and skipped notes in metronome-free recordings;
- Reference performance playback with dynamic metronome and score following, either in conventional score view, or piano roll view;

- Loudness graph visualization over the score;
- Overlayed graphs for pitch and loudness between reference and current performances in piano roll view.

6.3 Evaluation method

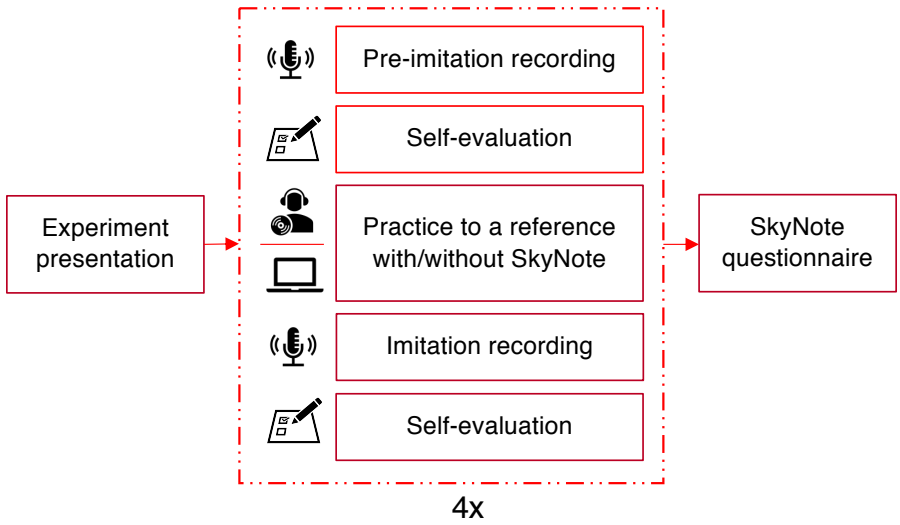


Figure 6.6: Structure of the experiment on expressive performance practice.

Participants of the study were instructed to fulfill a sequence of tasks structured as shown in the diagram of figure 6.6. After receiving an initial explanation followed by the collection of explicit participation consent and personal information, the musicians had to go through four rounds of a *performance imitation* exercise, in which their goal was to practice copying the expression of a reference performance of a given piece excerpt. The steps of each round of the exercise were the following:

Pre-imitation recording: the score to one of the four pieces, chosen randomly, is shown to the musician, and they are given up to 10 minutes to familiarize themselves with it. Once satisfied, the participant records their own interpretation of the piece.

The pieces selected for the exercise were chosen for having clear melodic lines and being short and technically simple. The scores are included in appendix A.

Self-evaluation: The participant is instructed to grade their previous performance in several criteria using a 7-point Likert scale from "low" (1) to "high" (7). The criteria are: "performance quality", "technical competence", "musicality", "intonation/note accuracy", "rhythmic accuracy", "tone quality", "dynamic control", "quality of articulation", and "room for improvement". The criteria list is deliberately extensive to provoke longer and deeper reflection.

Practice: In this step, the participant is given access to the reference recording they are supposed to imitate. For 2 of the 4 pieces, the imitation practice must be done only aurally, whereas for the remaining 2 pieces, the participant is instructed on how to use SkyNote for this purpose. The choice of which pieces should be imitated in each condition is randomized, but consecutive exercises always alternate the condition (imitation without SkyNote followed by imitation with SkyNote, and vice-versa). Before the first contact with the software, the participant is given an explanation on how to operate it by the accompanying researcher. Total practice time for each piece is again limited to 10 minutes, or whenever the musician is satisfied. Only one reference recording was used for each piece, regardless of the practice condition; however, the recordings used for pieces "Twinkle Twinkle Little Star" and "Manhã de Carnaval" featured only the violin, whereas "Frère Jacques" and "Greensleeves" had accompanying instruments.

Imitation recording: The musician records an imitation of the reference performance under the same conditions as the previous recording. It is important to note that no feedback from SkyNote or any other means are available at this point – the imitation is done by memory alone.

2nd self-evaluation: Another self-evaluation form, analogous to the previous, is presented. In addition to the previous criteria, participants are also inquired on: "difference from previous recording", "efficiency of your practice", "mental effort required for this exercise", and "physical effort required for this exercise". They are also encouraged to provide written comments justifying their evaluation.

After the completion of the imitation exercises, a final survey about SkyNote as a performance practice tool was conducted. All written explanations, questionnaires, and ran-

domizations were programmed as a web page to provide a consistent experience among participants.

6.4 Results and discussion

Six violinists took part in the system evaluation. Their background and experience ranged from beginner to professional, giving us insights on the perception of our setup from various perspectives. Table 6.1 serves as a summary of their profiles.

Participant	Years Playing/ Taking Lessons	Primary Genre	Musical Activity	Plays Reading*	Practices Expression †
P1	12/2	Classical	Amateur	●●●●○	●○○○○
P2	7/6	Traditional	Amateur	●●○○○	●●○○○
P3	5/2	Pop	Amateur	●●●○○	●●○○○
P4	3/3	Traditional	Amateur	●●●○○	●●●○○
P5	20/12	Classical	Professional	●●●●●	●●●●○
P6	10/10	Classical	Amateur	●●●○○	●○○○○

* Answer to question "How often do you play reading from a score?".

† Answer to question "How often do you practice musical expression?".

Table 6.1: Profiles of SkyNote evaluation participants.

A few correlations can be pointed out in the data from table 6.1. It is apparent that the emphasis on scores as the main medium for recording musical compositions in the classical repertoire leads musicians from that tradition to rely on reading more often while playing. Another relevant fact is that the participant who claims to practice musical expression the most is the only professional musician in the group. Finally, it is worth mentioning that the number of years of experience and study have proven to be good indicators of how easily each musician was able to complete the required tasks. Participants P2, P5, and P6 had virtually no difficulty learning the necessary melodies, and were able to approximate the interpretations of the references after practicing. P1, P3, and P4, on the other hand, had some degree of difficulty playing correctly or memorizing the melodies well enough to be able to fully focus on expression in the short time that was given, especially P4.

Although the number of participants is insufficient for a fully quantitative analysis, some aggregate responses of the self-evaluations help us better interpret their experiences and comments.

Mental effort

Several answers and remarks by the participants indicate that the imitation exercise was, at times, cognitively taxing:

> "Following the visual cues was very difficult as there are many. I can just concentrate in one. Either pitch curve (this one is very useful to follow timing), dynamics or score."
>
> P6, after practicing "Manhã de Carnaval" using SkyNote

We also observed that the initial practice time was often not enough for players to memorize the themes, forcing them to read from the score during their attempts to imitate the reference recording. Some participants also realized that this was hindering their performance:

> "This song was harder for me because I haven't memorized it, I kept reading while playing and that does not work for me."
>
> P4, after practicing "Greensleeves" using SkyNote (translated)

The ratings attributed to the item "mental effort required for this exercise" in the self-evaluation questionnaire (fig. 6.7) reflect these observations. In the rounds that included use of the software, the mean rating was higher than the in the remainder by a small margin, possibly a consequence of the excess of information and the learning process of the software itself. A positive observation, though, is that ratings for the fourth and last round of the exercise were, on average, lower than for the first, signaling a quick learning curve and even offsetting effects from fatigue, which were also mentioned:

> "I felt tired from the mental effort from the previous piece."
>
> P4, after the third round (translated)

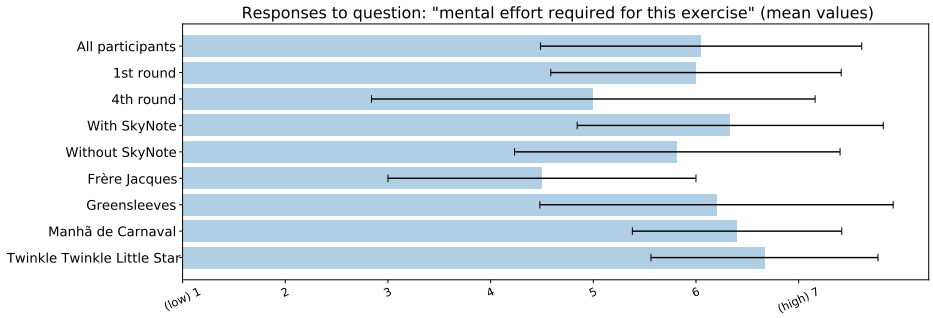


Figure 6.7: Breakdown of reported mental effort for the imitation exercise.

We conclude from these observations that reducing the number of simultaneous cognitive stimuli is important for effective use of visual feedback in music performance practice. Alternatives to achieve this include requiring the selection to visualize a single expressive feature, or a summary of features that represents them in a natural way, as with the interfaces proposed by Dixon et al. (2002) and Sadakata et al. (2008). An additional measure that could be encouraged is ensuring that real-time guided practice is proposed only for pieces that the student has fully committed to memory.

Performance context

Participants have also reported it to be easier to imitate the accompanied performances than the solo violin ones:

> "It was more comfortable to practice with more instruments."
>
> P3, after practicing "Greensleeves" using SkyNote (translated)

> "I found this one easier. I think because it is a "real" performance with accompaniment. There is a clear flow and intention. Even if I can't remember details I get the "intention" of the performance which allows me to perform fluently without thinking much on the details."
>
> P6, after practicing "Greensleeves" without SkyNote

In addition to the comments, the mean reported mental effort ratings for both accompanied pieces were slightly lower than for unaccompanied pieces, as shown in figure 6.7. As mentioned earlier, though, the quantitative observations are limited in their reliability due to the small number of participants.

Although very logical in nature, higher clarity of expression in ensembles than in solo performances is not an entirely obvious conclusion to be drawn *a priori*, but taking this effect into account can positively affect performance teaching even in a more traditional setup. It is interesting to note that this observation coincides with our hypotheses for participants' difficulties in ranking our synthesized performances of solo violin melodies in the experiment presented in chapter 4.

Perceptions and learning

Participants' performances during the imitation practice were markedly distinct from their original renditions. On some instances, particularly with more experienced players, this was simply a consequence of their different interpretations of the pieces, but there were also cases, especially during the first round, in which the players' original recordings lacked expressive intention, either caused by distraction from reading the score or an absent-minded attitude while performing.

> "Simply listening to the recording made me feel the tempo better and relax the wrist, like when a teacher shows you how to play a piece."
>
> P4, after the first round

Many more elements influenced the self-evaluation ratings given by participants. Some expressed a distaste for the references they were asked to imitate, which impacted their motivation. Difficulty to memorize the details of the reference interpretation was a frequent complaint:

> "I can listen and detect many features, but I have difficulties trying to remember them." P6, after the first round

With all these elements in mind, we can better interpret the mean ratings in responses by participants, presented in figure 6.8. When asked to rate the quality of their performances before and after the imitation exercise, participants did not report higher quality for the later recording, though they were aware of the differences between them. Results from the rounds using feedback from SkyNote were very similar to the rounds without it. The small differences observed for practice efficiency and perception of room for improvement are consistent with our vision that SkyNote is successful in making musicians more aware of the expressive possibilities in their performance, but the short experience with the software was insufficient to help them incorporate such techniques in their playing. We would expect, therefore, that these differences be confirmed if testing were conducted in larger groups.

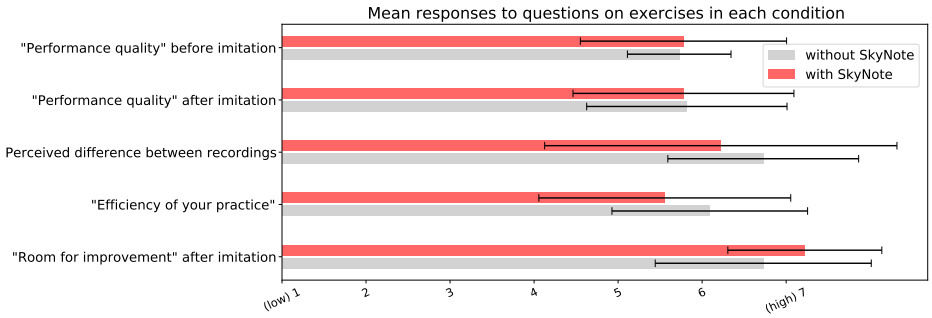


Figure 6.8: Contrasts between survey responses when practicing with or without the software.

Given all factors discussed, we may argue that the imitation exercise as it was realized, despite its advantage of being a very tangible task, can also detract from the long-term learning experience that would be possible from the mere exposure to reference performances.

It was possible to notice that while novice players would pay close attention to the software feedback for cues that could help them parse the expression on reference performances, experienced players would often do the opposite, actively looking away to put more focus on their listening skills. This illustrates how the inclusion of new tools represents a disruption in the study process that experienced players have already established, and their adoption, if deemed useful by them, would follow a different process than for novice players. In accordance with this reasoning, P5, the professional player, remarks:

> "I believe it would be good to test the program with children or youngsters beginning violin studies to know if it is useful for them, to discover whether it really helps the study of the violin in this level."
>
> P5 (translated)

Opinions on SkyNote

Feedback from participants helped us reach a deeper understanding about the quality of SkyNote's design in terms of two main aspects: its accuracy – whether the information conveyed is correct and appropriate – and its acceptance – whether musicians are stimulated to use it and find value in its adoption. The responses to the post-experiment survey are summarized by figure 6.9. Ratings were predominantly positive in all respects, but a highlight is that mean ratings were higher for questions related to interest and long-term use of the system (e.g: "To what degree was SkyNote something you would use again?", "To what degree was SkyNote something you would recommend?") than for those related to its immediate impact in the proposed task (e.g.: "To what degree did SkyNote improve your performance?").

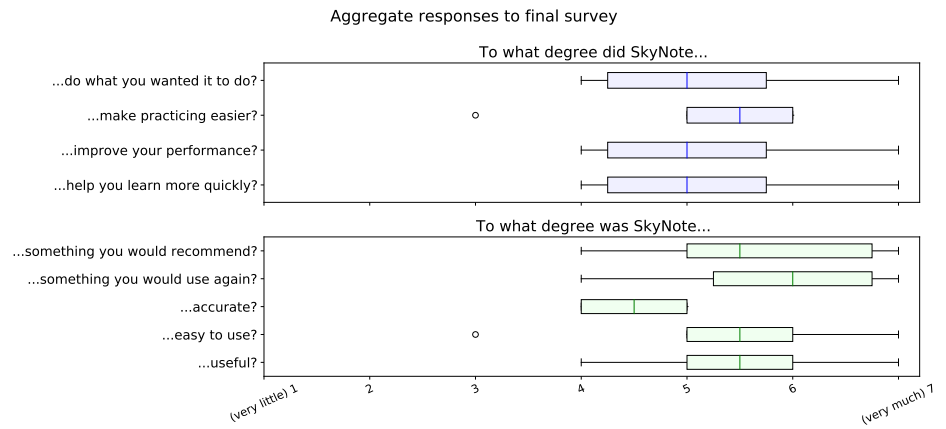


Figure 6.9: Distributions of answers from the survey on SkyNote.

Looking deeper into the comments of participants who gave the lowest ratings reveals the issues they encountered. P4, who replied to the question "To what degree did SkyNote make practicing easier" with 3 out of 7, included the following comment as SkyNote's

greatest weakness or limitation:

> "I haven't used the other functionalities [besides note grading, synthesized, and accompanied audio playback] because of the songs I had to play. It would be nice if at least one easy one could be played with the software."
>
> P4 (translated)

Indeed, as we had mentioned, P4 found most required pieces technically difficult, so we might interpret that it was too early, in the state of their practice, to place so much emphasis on expressive performance concerns. This is reinforced by the participants' favorite features – note grading, synthesized playback of scores and reference performance score-followed playback – which were part of the original set of features in SkyNote, targeting novice players. This is still a positive result for our evaluation, since these features were only made available for the practice of performances due to our audio-to-score alignment and dynamic tempo implementations.

The other lowest rating was another 3 out of 7 given by P6 on the question "To what degree was SkyNote easy to use?". Their response to the inquiry on SkyNotes' weaknesses reads:

> "Too much information at the same time. Can just concentrate on one thing. Needs more user-friendlyness." P6

This criticism echoes our discussion on the high mental effort demanded by the exercise. Our conclusion on the topic is also shared by P6, as they stated in the following "suggestion for improvement":

> "Dedicated exercises and visualization for specific skills. Playback/recording and off-line visualizations for retrospective analysis is the best [practice method]." P6

On the adequacy of feedback-related software features, different participants showed preference for different visualizations. P1 highlighted the dynamics visualization over

the score as the most helpful in their opinion, whereas P3 and P5 chose the piano roll view. The others did not single out one of the two views as most helpful. This variation in opinions is an evidence that neither is strictly superior, but rather that this choice depends on preference and occasion. Both classical musicians proposed some form of score annotation as an improvement:

> "Having a tool that allows you to make notes on the score, or an option of comments about the exercise, to make it easier to remember the details to be improved." P5, on how to improve SkyNote feedback

> "If you change the note lengths in the staff based on the expected duration of the note it will introduce some visual feedback about rhythm." P1, on the same topic

The inclusion of clearer rhythmic feedback when using the score view was also suggested by P2. A general purpose annotation, as P5 suggests, would be a technically simple solution for having information on complex expressive features. Watching participants practice the exercises, one would notice that most of the time was dedicated to imitating articulations and vibrato, which involve techniques very specific to the violin, and which are only reported indirectly by the software, via the pitch curve. Still, this was considered useful, as illustrated in the following comment by P3:

> "The manner of giving feedback on vibrato during the imitation was pleasant to view, but it took me some effort to put it to use, because there were too many stimuli." P3, on SkyNote weaknesses and limitations

Finally, the feedback on the accuracy of technical implementations suggests that the experience could be improved with careful filtering of the pitch extraction algorithm. P2 remarked on its visualization:

> "I would have liked seeing note durations more clearly in the piano roll, because the pitch detection wasn't fully precise." P2 (translated)

A filtered pitch detection would also have a positive influence in the outcome of the audio-to-score alignment in situations where audio capture is not optimal. Precise alignment is key for proper rhythmic feedback and note corrections, which lend support especially to students struggling with a certain piece. This was an issue to P4 in one of the exercises:

> "I was confused by the current note indicator, it seemed to go out of synchrony with the audio at times, and I got a little stressed trying to catch up to it." P4 (translated)

Besides the rich observations that we were able to report from the feedback we received, our conclusions from this experience also support the idea that the best strategy for studying the effects of a technology-enhanced learning scenario for music, and especially music expression, is a long-term accompaniment of participants with the help of their own tutors. This would be a favorable setting for learning about skill retention, for mitigating biases caused by issues related to the software learning curve and fatigue from long sessions, and especially to understand the benefits of this study methodology after the musician learns to incorporate the software feedback in their own practice routine.

The evaluation process, from our perspective, has been successful in avoiding skeptical opinions towards technological tools in a musical environment – a recurring occurrence in research, as we pointed out in chapter 2 – due to the concrete nature of the proposal. Instead, the process has been instrumental in evolving our ideas about the suitability of technology-enhanced settings for learning expression and the proper conditions to do so. We hope that these findings can encourage further exploration of technology in music classrooms, whether systematically in academia, or individually by the innovative and creative music teachers everywhere.

> "It made me reevaluate my own expressivity comparing it to others over the dynamics that are possible with the violin, and that helps me broaden this creative side." P3 (translated)

Chapter 7

Conclusions

Over the previous chapters we have approached the computer modeling of expressive performance actions by various angles, proposing models that prioritize simplicity, then flexibility, and finally, accuracy. We also explored the type of music education scenario that could leverage such models to offer learning musicians more options of creative stimuli in the development of their personal sensibilities and expressive styles.

The fast evolution of the machine-learning field during recent years has made it difficult to systematically evaluate the quality gains obtained from the application of novel methods. Beyond proposing one particular computer model best suited for one particular application and subject to the availability of a specific bundle of musical information, the research presented in this thesis should be viewed as a broken down report on the impact of each design choice in modeling musical expression at the time of this writing, thus emphasizing the possibilities and the limitations with current techniques and resources.

Considering the future of expressive performance modeling, one might start by addressing the problems that we have come across in our analyses. The prime example is the quality of datasets. Doubtlessly, the heterogeneous recording conditions, unreliable note onset alignments, and the mixture of musical styles were all factors that negatively impacted model learning. Improving automatic methods for ensuring consistency might be a strategy for overcoming these issues without sacrificing orders of magnitude in dataset sizes. Drawing inspiration from the neural translation field once again, the

methodology made famous by the BERT model (Devlin et al., 2019), consisting of pre-training large-scale models on very large (but simpler) datasets followed by fine-tuning in application-specific, smaller datasets, seems promising. Not only could it solve the dataset consistency problems, this approach also allows a creative personalization of the training dataset, as we encouraged with our proposals from chapters 3 and 4. Following the research trend that we pointed out in our discussion of deep-learning applications, the automatic music composition field has already begun to explore this pipeline. Huang and Yang (2020) not only combine pre-training and fine-tuning, as they also propose a new musical encoding to address the issues from the Music Transformer encoding that we identified. It would be interesting to analyze their new REMI encoding against our own in a similar architecture.

Numerically, the performances generated by state-of-the-art CSEMP are approaching the limits represented by the variance which exists between performances of the same piece by different musicians, so the improvement of models necessarily goes through refining the formulation of the problem, that is, understanding what questions we should ask about what makes performances unique and pleasing in a way that lends itself to other systematic yet inspiring ways to look at music expression.

A consistent observation across all our experiments that warrants commenting is the complexity of violin expression from a technical standpoint. A variety of EPAs work in tandem during a violin performance, many of which are challenging to quantify consistently and automatically via music information retrieval techniques. CSEMP literature has barely scratched the surface of this topic, and much can be learned from investigating how interdependent these complex EPAs are, and what kind of performance learning is transferrable between musical instruments.

Little was said in previous chapters about the performance of ensembles and how important the interaction between musicians in this scenario is. Nevertheless, our CSEMP have been trained on performances of ensembles, and the patterns they learned certainly include information of that nature. In particular, clever application of our sequence models of chapter 5 could generate performance signals for one musician conditioned on the expression of their bandmates. This could be achieved by the simple substitution of model outputs by real performance signals of the existing musicians in subsequent steps of the generation, in a process analogous to the operation of the Flow Machines

model of composition (Pachet et al., 2020). The analysis of performances produced in that manner is a planned continuation of this work.

With respect to the pedagogical utility of CSEMP, as we have highlighted before, generated performances can stimulate the critical analysis of expression and fruitful discussions between musicians. Its impact, though, can be even deeper. Technology has always profoundly influenced how we perceive and make music, from the evolution of tuning changing our perceptions of tonality (Michèle Duguay, 2016) to the role of music recording in the birth of modern musical genres (Borthwick and Moy, 2004). If computer systems can improve our understanding of what happens in a music performance, they will surely be part of the next aesthetic leap in the musical arts.

Bibliography

D. W. Aha. *Lazy Learning*. Springer, Dordrecht, first edition, June 2013. ISBN 978-94-017-2053-3.

V. Akkermans, F. Font, J. Funollet, B. de Jong, G. Roma, S. Togias, and X. Serra. Freesound 2: An improved platform for sharing audio clips. In *12th International Society for Music Information Retrieval (ISMIR)*, Miami, Florida (USA), 2011.

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, California (USA), 2015.

N. K. Baker, M. Paddison, and R. Scruton. Expression. *Grove Music Online. Oxford Music Online*, 1, 2001. doi:10.1093/gmo/9781561592630.article.09138.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, Mar. 1994. ISSN 1941-0093. doi:10.1109/72.279181.

Y. Bengio, Y. Lecun, and G. Hinton. Deep learning for AI. *Communications of the ACM*, 64(7):58–65, July 2021. ISSN 0001-0782, 1557-7317. doi:10.1145/3448250.

C. Bonastre and R. Timmers. Comparison of beliefs about teaching and learning of emotional expression in music performance between Spanish and English HE students of music. *Psychology of Music*, 49(1):108–123, Jan. 2021. ISSN 0305-7356. doi:10.1177/0305735619842366.

S. Borthwick and R. Moy. *Popular Music Genres: An Introduction*. Routledge, New

York, 2004. ISBN 978-1-315-02456-1. doi:10.4324/9781315024561.

R. Bresin and A. Friberg. Evaluation of Computer Systems for Expressive Music Performance. In *Guide to Computing for Expressive Music Performance*, pages 181–203. Springer London, London, 2013. doi:10.1007/978-1-4471-4123-5_7.

R. Bresin, A. Friberg, and J. Sundberg. Director Musices: The KTH Performance Rules System. In *Proceedings of SIGMUS46*, pages 43–48, Kyoto, Japan, 2002. Information Processing Society of Japan.

J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 07(04):669–688, Aug. 1993. ISSN 0218-0014. doi:10.1142/S0218001493000339.

E. Cambouropoulos. The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing. In *Proceedings of the 2001 International Computer Music Conference (ICMC)*, pages 17–22, Havana, Cuba, 2001.

C. E. Cancino-Chacón and M. Grachten. The basis mixer: A computational romantic pianist. In *Proceedings of the Late Breaking/ Demo Session, 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, New York, NY, USA, 2016.

C. E. Cancino-Chacón, T. Gadermaier, G. Widmer, and M. Grachten. An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. *Machine Learning*, 106(6):887–909, 2017. ISSN 15730565. doi:10.1007/s10994-017-5631-y.

C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer. Computational Models of Expressive Music Performance: A Comprehensive and Critical Review. *Frontiers in Digital Humanities*, 5, 2018. ISSN 2297-2668. doi:10.3389/fdigh.2018.00025.

K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv:1409.1259 [cs, stat]*, Oct. 2014.

M. Clynes. Microstructural musical linguistics: Composers' pulses are liked most by the best musicians. *Cognition*, 55(3):269–310, June 1995. ISSN 00100277. doi:10.1016/0010-0277(94)00650-A.

A. Cont. ANTESCOFO: Anticipatory Synchronization and Control of Interactive

Parameters in Computer Music. In *Proceedings of the 2008 International Computer Music Conference*, pages 33–40, Belfast, Northern Ireland, Aug. 2008.

D. Cooke. *The Language of Music*. Oxford University Press, 1989. ISBN 978-0-19-816180-6.

A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, Jan. 2018. ISSN 1558-0792. doi:10.1109/MSP.2017.2765202.

R. B. Dannenberg. The Interpretation of MIDI Velocity. In *Proceedings of the 2006 International Computer Music Conference*, pages 193–196, New Orleans, Louisiana (USA), 2006.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota (USA), June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

S. Dixon, W. Goebl, and G. Widmer. The Performance Worm: Real Time Visualisation of Expression based on Langner's Tempo-Loudness Animation. In *Proceedings of the 2002 International Computer Music Conference (ICMC)*, page 4, Gothenburg, Sweden, 2002.

EBU TC Committee. Tech 3341: Loudness metering: 'EBU mode' metering to supplement EBU R 128 loudness normalization. Technical report, EBU, Geneva, 2016.

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

T. Eerola and P. Toiviainen. MIDI Toolbox: MATLAB Tools for Music Research, 2004.

S. Flossmann, W. Goebl, M. Grachten, and G. Widmer. The Magaloff Project: An Interim Report. *Journal of New Music Research*, 39:363–377, Dec. 2010. doi:10.1080/09298215.2010.523469.

S. Flossmann, M. Grachten, and G. Widmer. Expressive Performance Rendering with Probabilistic Models. In A. Kirke and E. R. Miranda, editors, *Guide to Computing*

for Expressive Music Performance, pages 75–98. Springer, London, 2013. ISBN 978-1-4471-4123-5. doi:10.1007/978-1-4471-4123-5_3.

E. Frank, M. A. Hall, and I. H. Witten. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. Morgan Kaufmann, 2016.

F. N. Fritsch and R. E. Carlson. Monotone Piecewise Cubic Interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, Apr. 1980. ISSN 0036-1429, 1095-7170. doi:10.1137/0717021.

A. Gabrielsson. Music Performance Research at the Millennium. *Psychology of Music*, 31(3):221–272, 2003. ISSN 0305-7356. doi:10.1177/03057356030313002.

A. Gabrielsson, I. Bengtsson, and B. Gabrielsson. Performance of musical rhythm in 3/4 and 6/8 meter. *Scandinavian Journal of Psychology*, 24(1):193–213, 1983. ISSN 14679450. doi:10.1111/j.1467-9450.1983.tb00491.x.

T. Gadermaier, M. Grachten, and C. E. C. Chacón. Basis-Function Modeling of Loudness Variations in Ensemble Performance. In *2nd International Conference on New Music Concepts (ICNMC)*, Treviso, Italy, 2016.

S. I. Giraldo and R. Ramirez. A Machine Learning Approach to Discover Rules for Expressive Performance Actions in Jazz Guitar Music. *Frontiers in Psychology*, 7(DEC): 1965, Dec. 2016. ISSN 1664-1078. doi:10.3389/fpsyg.2016.01965.

R. O. Gjerdingen and E. Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*, volume 50. University of Chicago Press, 2006. ISBN 0-226-56842-3. doi:10.2307/898334.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, Sardinia, Italy, Mar. 2010. JMLR Workshop and Conference Proceedings.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, Ft. Lauderdale, Florida (USA), 2011. JMLR Workshop and Conference Proceedings.

M. Grachten and G. Widmer. Linear Basis Models for Prediction and Analysis of Musical Expression. *Journal of New Music Research*, 41(4):311–322, Dec. 2012. ISSN

0929-8215. doi:10.1080/09298215.2012.731071.

L. F. Hamond, G. Welch, and E. Himonides. The Pedagogical Use of Visual Feedback for Enhancing Dynamics in Higher Education Piano Learning and Performance. *OPUS*, 25(3):581–601, Dec. 2019.

C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *Seventh International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana (USA), 2019.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov. 2012. ISSN 1558-0792. doi:10.1109/MSP.2012.2205597.

S. Hochreiter. Recurrent neural net learning and vanishing gradient. *International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems*, 6(2):107–116, 1998.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

D. R. Hofstadter. The Ineradicable Eliza Effect and Its Dangers. In *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic books, 1995.

C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music Transformer. *arXiv:1809.04281 [cs, eess, stat]*, Dec. 2018.

Y.-S. Huang and Y.-H. Yang. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pages 1180–1188, New York, NY, USA, Oct. 2020. Association for Computing Machinery. ISBN 978-1-4503-7988-5. doi:10.1145/3394171.3413671.

D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam. VirtuosoNet: A Hierarchical RNN-based system for modeling expressive piano performance. In *20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019a.

D. Jeong, T. Kwon, Y. Kim, and J. Nam. Graph Neural Network for Music Score Data and Modeling Expressive Piano Performance. In *International Conference on Machine Learning (MLR)*, pages 3060–3070, Long Beach, California (USA), 2019b.

S. Ji, J. Luo, and X. Yang. A Comprehensive Survey on Deep Music Generation: Multi-level Representations, Algorithms, Evaluations, and Future Directions. *arXiv:2011.06801 [cs, eess]*, Nov. 2020.

J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa. Transformer VAE: A Hierarchical Model for Structure-Aware and Interpretable Music Representation Learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 516–520, Barcelona, Spain (online), May 2020. doi:10.1109/ICASSP40776.2020.9054554.

Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. *arXiv:1611.05148 [cs]*, June 2017.

S. Jordà. Instruments and Players: Some Thoughts on Digital Lutherie. *Journal of New Music Research*, 33(3):321–341, Sept. 2004. ISSN 0929-8215, 1744-5027. doi:10.1080/09298210420003178 86.

P. Juslin, J. Karlsson, E. Lindström, A. Friberg, and E. Schoonderwaldt. Play It Again With Feeling: Computer Feedback in Musical Communication of Emotions. *Journal of experimental psychology. Applied*, 12:79–95, July 2006. doi:10.1037/1076-898X.12.2.79.

J. Karlsson and P. N. Juslin. Musical expression: An observational study of instrumental teaching. *Psychology of Music*, 36(3):309–334, July 2008. ISSN 0305-7356, 1741-3087. doi:10.1177/0305735607086040.

J. Karlsson, S. Liljeström, and P. N. Juslin. Teaching musical expression: Effects of production and delivery of feedback by teacher vs. computer on rated feedback quality. *Music Education Research*, 11(2):175–191, June 2009. ISSN 1461-3808. doi:10.1080/14613800902924532.

H. Katayose, M. Hashida, G. De Poli, and K. Hirata. On Evaluating Systems for Generating Expressive Music Performance: The Rencon Experience. *Journal of New Music Research*, 41(4):299–310, 2012. ISSN 0929-8215. doi:10.1080/09298215.2012.745579.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint*

arXiv:1412.6980, 2014.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014.

A. Kirke and E. R. Miranda. A survey of computer systems for expressive music performance. *ACM Computing Surveys*, 42(1):1–41, Dec. 2009. ISSN 03600300. doi:10.1145/1592451.1592454.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2012. ISSN 0001-0782, 1557-7317. doi:10.1145/3065386.

P. Laukka. Instrumental music teachers' views on expressivity: A report from music conservatoires. *Music Education Research*, 6(1):45–56, Mar. 2004. ISSN 1461-3808. doi:10.1080/1461380032000182821.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi:10.1038/nature14539.

Leopold Auer. *Violin Playing As I Teach It*. 1920.

Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated Graph Sequence Neural Networks. *arXiv:1511.05493 [cs, stat]*, Sept. 2017.

K. A. Lim and C. Raphael. InTune: A System to Support an Instrumentalist's Visualization of Intonation. *Computer Music Journal*, 34(3):45–55, Sept. 2010. ISSN 0148-9267, 1531-5169. doi:10.1162/COMJ_a_00005.

E. Lindström, P. Juslin, R. Bresin, and A. Williamon. "Expressivity comes from within your soul": A questionnaire study of music students' perspectives on expressivity. *Research Studies in Music Education*, 20:23–47, June 2003a. doi:10.1177/1321103X030200010201.

E. Lindström, P. N. Juslin, R. Bresin, and A. Williamon. "Expressivity comes from within your soul": A questionnaire study of music students' perspectives on expressivity. *Research Studies in Music Education*, 20(1):23–47, 2003b. ISSN 1321-103X. doi:10.1177/1321103X030200010201.

T. Lisboa, A. Williamon, M. Zicari, and H. Eiholzer. Mastery through imitation. In

European Society for the Cognitive Sciences of Music Conference (ESCOM), Liège, Belgium, 2002.

S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson. Changing musical emotion: A computational rule system for modifying score and performance. *Computer Music Journal*, 34(1):41–64, Mar. 2010. ISSN 01489267. doi:10.1162/comj.2010.34.1.41.

E. Maestre. *Modeling Instrumental Gestures : An Analysis / Synthesis Framework for Violin Bowing*. PhD thesis, 2009.

I. Malik and C. H. Ek. Neural Translation of Musical Style. *arXiv:1708.03535 [cs]*, Aug. 2017.

M. Marchini, R. Ramirez, P. Papiotis, and E. Maestre. The Sense of Ensemble: A Machine Learning Approach to Expressive Performance Modelling in String Quartets. *Journal of New Music Research*, 43(3):303–317, July 2014. ISSN 0929-8215. doi:10.1080/09298215.2014.922999.

M. Mauch and S. Dixon. pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, volume 1, pages 659–663, 2014. ISBN 978-1-4799-2893-4.

M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, France, 2015.

N. P. McAngus Todd. The dynamics of dynamics: A model of musical expression. *The Journal of the Acoustical Society of America*, 91(6):3540–3550, 1992. ISSN 0001-4966. doi:10.1121/1.402843.

H. Meissner. Instrumental teachers' instructional strategies for facilitating children's learning of expressive music performance: An exploratory study. *International Journal of Music Education*, 35(1):118–135, Feb. 2017. ISSN 0255-7614, 1744-795X. doi:10.1177/0255761416643850.

H. Meissner and R. Timmers. Teaching young musicians expressive performance: An experimental study. *Music Education Research*, 21(1):20–39, Jan. 2019. ISSN 1461-3808. doi:10.1080/14613808.2018.1465031.

H. Meissner, R. Timmers, and S. E. Pitts. 'Just notes': Young musicians' perspectives on learning expressive performance. *Research Studies in Music Education*, page 1321103X19899171, Sept. 2020. ISSN 1321-103X. doi:10.1177/1321103X19899171.

Michèle Duguay. *The Influence of Unequal Temperament on Chopin's Piano Works*. PhD thesis, Schulich School of Music, McGill University, Montreal, Canada, 2016.

S. Moulieras and F. Pachet. Maximum entropy models for generation of expressive music. *arXiv:1610.03606 [cs]*, Oct. 2016.

M. Müller. Dynamic Time Warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer, Berlin, Heidelberg, 2007. ISBN 978-3-540-74048-3. doi:10.1007/978-3-540-74048-3_4.

T. Nakamura, E. Nakamura, and S. Sagayama. Acoustic Score Following To Musical Performance With Errors and Arbitrary Repeats and Skips. In *Proceedings of the Sound and Music Computing Conference 2013 (SMC2013)*, pages 299–304, Stockholm, Sweden, 2013. ISBN 978-3-8325-3472-1.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, Mar. 1970. ISSN 00222836. doi:10.1016/0022-2836(70)90057-4.

S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, 2018. ISSN 1433-3058. doi:10.1007/s00521-018-3758-9.

F. J. M. Ortega, S. Giraldo, and R. Ramírez. Bowing modeling for violin students assistance. In *Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education*, pages 60–62, Glasgow, UK, 2017a. Association for Computing Machinery. ISBN 978-1-4503-5557-5. doi:10.1145/3139513.3139525.

F. J. M. Ortega, S. Giraldo, and R. Ramírez. Phrase-level modeling of expression in violin performances. In *10th International Workshop on Machine Learning and Music*, pages 49–54, Barcelona, 2017b. Machine Learning and Music (MML).

F. J. M. Ortega, S. Giraldo, A. A. Pérez Carrillo, and R. Ramírez. Phrase-Level modeling of expression in violin performances. *Frontiers in Psychology*, 10:776, 2019a. ISSN 1664-1078. doi:10.3389/fpsyg.2019.00776.

F. J. M. Ortega, A. A. Pérez Carrillo, and R. Ramírez. Predicting dynamics in violin pieces with features from melodic motifs. In *Joint European Conference on Machine*

Learning and Knowledge Discovery in Databases (ECML PKDD), pages 517–523, Würzburg, Germany, 2019b. Springer, Cham. doi:10.1007/978-3-030-43887-6_46.

F. Pachet, P. Roy, and B. Carré. Assisted music creation with Flow Machines: Towards new categories of new. In *Handbook of Artificial Intelligence for Music*. Springer, 2020.

C. Palmer. Music performance. *Annual review of psychology*, 48(1):115–138, 1997. doi:10.1146/annurev.psych.48.1.115.

P. Papiotis, M. Marchini, A. Perez-Carrillo, and E. Maestre. Measuring ensemble interdependence in a string quartet through analysis of multidimensional performance data. *Frontiers in Psychology*, 5(AUG):963, Sept. 2014. ISSN 16641078. doi:10.3389/fpsyg.2014.00963.

B. Patterson. Musical Dynamics. *Scientific American*, 231(5):78–95, 1974. ISSN 0036-8733.

C. Peiper, D. Warden, and G. Garnett. An Interface for Real-time Classification of Articulations Produced by Violin Bowing. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME'03)*, pages 192–196, 2003.

R. Ramirez, A. Hazan, E. Maestre, and X. Serra. A Genetic Rule-Based Model of Expressive Performance for Jazz Saxophone. *Computer Music Journal*, 32(1):38–50, 2008. ISSN 01489267. doi:10.1162/comj.2008.32.1.38.

R. Ramirez, C. Canepa, S. Ghisio, K. Kolykhalova, M. Mancini, E. Volta, G. Volpe, S. Giraldo, O. Mayor, A. Perez, G. Waddell, and A. Williamon. Enhancing Music Learning with Smart Technologies. In *Proceedings of the 5th International Conference on Movement and Computing*, MOCO '18, pages 1–4, New York, NY, USA, June 2018. Association for Computing Machinery. ISBN 978-1-4503-6504-8. doi:10.1145/3212721.3212886.

B. H. Repp. Some empirical observations on sound level properties of recorded piano tones. *Journal of the Acoustical Society of America (JASA)*, 93(2):1136–44, Feb. 1993. ISSN 0001-4966. doi:10.1121/1.405561.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986. ISSN 1476-4687. doi:10.1038/323533a0.

M. Sadakata, D. Hoppe, A. Brandmeyer, R. Timmers, and P. Desain. Real-Time Vi-

sual Feedback for Learning to Perform Short Rhythms with Expressive Variations in Timing and Loudness. *Journal of New Music Research*, 37(3):207–220, Sept. 2008. ISSN 0929-8215. doi:10.1080/09298210802322401.

C. E. Seashore. The Psychology of Music. *Music Educators Journal*, 25(3):23–23, Dec. 1938. ISSN 0027-4321. doi:10.2307/3385515.

X. Serra and J. Smith. Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition. *Computer Music Journal*, 14(4):12–24, 1990. ISSN 0148-9267. doi:10.2307/3680788.

I. Simon and S. Oore. Performance RNN: Generating music with expressive timing and dynamics, 2017.

S. W. Smoliar, J. A. Waterworth, and P. R. Kellock. pianoFORTE: A System for Piano Education Beyond Notation Literacy. In *Proceedings of the Third ACM International Conference on Multimedia - MULTIMEDIA '95*, pages 457–465, 1995. ISBN 0-89791-751-0. doi:10.1145/217279.215310.

K. Sohn, H. Lee, and X. Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems 28*, Montréal, Canada, 2015.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. ISSN 1533-7928.

D. R. Stammen and B. Pennycook. Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the 19th International Computer Music Conference (ICMC)*, pages 232–5, Tokio, Japan, 1993.

J. Sundberg, A. Askenfelt, and L. Frydén. Musical performance: A synthesis-by-rule approach. *Computer Music Journal*, 7(1):37–43, 1983. ISSN 01489267, 15315169.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, Montréal, Canada, 2014.

H. H. Tan, Y.-J. Luo, and D. Herremans. Generative Modelling for Controllable Audio Synthesis of Expressive Piano Performance. *arXiv:2006.09833 [cs, eess]*, July 2020.

D. Temperley. What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algo-

rithm Reconsidered. *Music Perception: An Interdisciplinary Journal*, 17(1):65–100, Oct. 1999. ISSN 07307829. doi:10.2307/40285812.

J. Thickstun, Z. Harchaoui, and S. Kakade. Learning Features of Music from Scratch. *arXiv:1611.09827 [cs, stat]*, Apr. 2017.

A. Tobudic and G. Widmer. Relational IBL in music with a new structural similarity measure. In *Proceedings of the 13th International Conference on Inductive Logic Programming*, pages 365–382, 2003.

A. Tobudic and G. Widmer. Relational IBL in classical music. *Machine Learning*, 64 (1-3):5–24, Sept. 2006. ISSN 0885-6125, 1573-0565. doi:10.1007/s10994-006-8260-4.

N. P. M. Todd. A model of expressive timing in tonal music. *Music Perception: An Interdisciplinary Journal*, 3(1):33–57, 1985. ISSN 07307829, 15338312.

J. Van Der Linden, E. Schoonderwaldt, J. Bird, and R. Johnson. MusicJacket - Combining motion capture and vibrotactile feedback to teach violin bowing. In *IEEE Transactions on Instrumentation and Measurement*, volume 60, pages 104–113, Jan. 2011. ISBN 0018-9456 VO - 60. doi:10.1109/TIM.2010.2065770.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, California (USA), 2017.

G. Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, June 2003. doi:10.1016/S0004-3702(03)00016-X.

G. Widmer and W. Goebl. Computational Models of Expressive Music Performance: The State of the Art. *Journal of New Music Research*, 33(3):203–216, 2004. ISSN 0929-8215. doi:10.1080/0929821042000317804.

G. Widmer and A. Tobudic. Playing Mozart by Analogy: Learning Multi-level Timing and Dynamics Strategies. *Journal of New Music Research*, 32(3):259–268, Sept. 2003. ISSN 0929-8215. doi:10.1076/jnmr.32.3.259.16860.

R. H. Woody. The Effect of Various Instructional Conditions on Expressive Music Performance. *Journal of Research in Music Education*, 54(1):21–36, Jan. 2006. ISSN 0022-4294. doi:10.1177/002242940605400103.

L. M. Zbikowski. *Foundations of Musical Grammar*. Oxford University Press, 2017.

ISBN 978-0-19-065363-7.

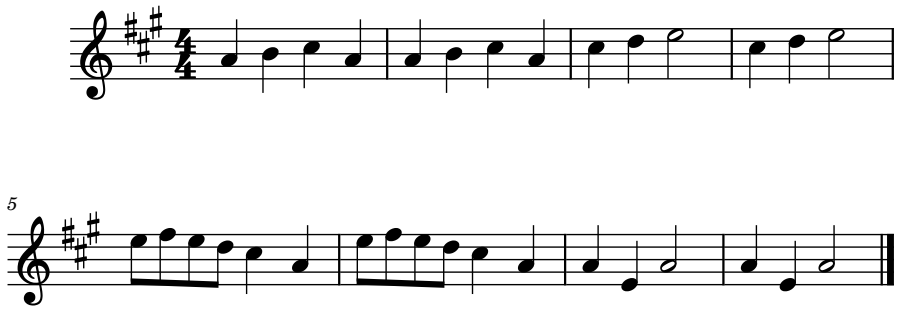
N. Zhang. Learning Adversarial Transformer for Symbolic Music Generation. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–10, 2020. ISSN 2162-2388. doi:10.1109/TNNLS.2020.2990746.

Appendix **A**

Imitation Exercise Scores

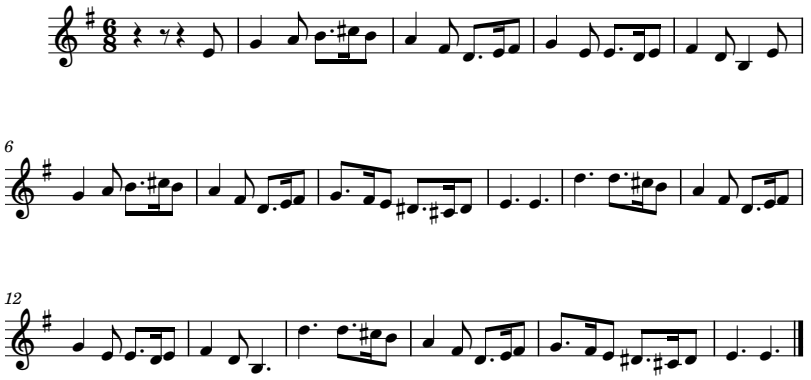
Frère Jacques

Traditional



Greensleeves

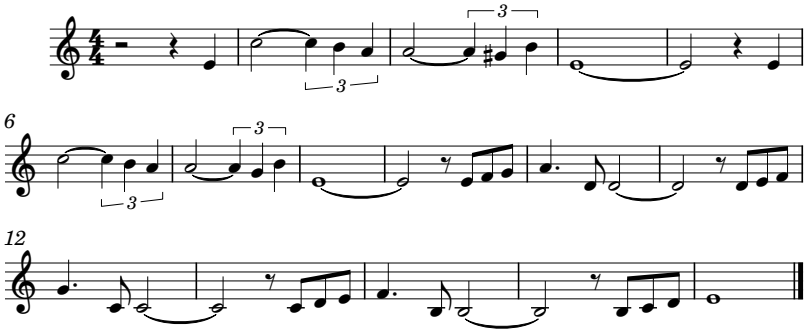
Traditional



Manhã de Carnaval

(from Black Orpheus)

Luís Bonfá



Twinkle Tinkle Little Star

Traditional