Generative, Discriminative, and Hybrid Approaches to Audio-to-Score Automatic Singing Transcription

Ryo NISHIKIMI

Abstract

This thesis describes audio-to-score automatic singing transcription (AST) methods that estimate a human-readable musical score of the sung melody from a music signal, where each note is represented by a semitone-level pitch (MIDI note number) and a note value (integer multiples of the tatum interval). The musical score is the most common format for describing, archiving, and distributing music, and audio-to-score AST plays a key role in deeper understanding of popular music including vocal parts.

To realize audio-to-score AST robust against the variation of singing voice, we should take three requirements into account. First, an acoustic model is required for describing the F0 and temporal deviations from musical scores. Second, to avoid the musically-unnatural estimate of musical notes, a language model is required for describing grammatical structures of musical scores. Third, to avoid the error propagation caused by the cascading audio-to-F0-to-score approach, musical notes should be estimated directly from music audio signals.

This thesis takes a principled approach to audio-to-score AST based on the integration of language and acoustic models. Because each model can be formulated in either a generative or discriminative manner, we propose generative, discriminative, and hybrid unified models, each of which consists of language and acoustic models. The key feature common to these unified models is that the most likely note sequence is estimated from a vocal F0 trajectory or spectrogram at once, while considering both the musical naturalness of notes and their fitness to the trajectory or spectrogram.

In Chapter 3, we take a *generative* approach based on a hierarchical hidden semi-Markov model (HSMM) of a vocal F0 trajectory that integrates a *generative*

language model describing the transitions of local keys and the rhythms of notes with a *generative* semi-acoustic model describing the time-frequency fluctuation of the trajectory. We experimentally show that the language model with a prior learned from existing scores improves the performance of AST.

In Chapter 4, we take a *hybrid* approach based on an HSMM of a vocal spectrogram that integrates a pretrained *generative* language model similar to that in Chapter 3 with a *discriminative* acoustic model based on a convolutional recurrent neural network (CRNN) trained in a supervised manner for predicting the posterior probabilities of note pitches and onsets from the spectrogram. We experimentally show that the CRNN-HSMM achieved state-of-the-art performance thanks to the combination of the grammatical knowledge about musical notes and the expressive power of the CRNN.

In Chapters 5, we take a *discriminative* approach based on a standard attention-based encoder-decoder model that uses a frame-level *discriminative* encoder and a *note-level discriminative* decoder for directly estimating musical notes (pitches and note values) from a vocal spectrogram. To make use of fewer aligned paired data for learning monotonic audio-to-score alignment, we propose a loss function for an attention matrix. We experimentally show the effectiveness of the attention loss and the strength and weakness of the model in estimating instantaneous and temporal attributes (*e.g.*, pitches and note values), respectively.

In Chapter 6, we propose an alternative encoder-decoder model consisting of a frame-level *discriminative* encoder and a *tatum-level discriminative* decoder for directly estimating sequences of pitches and binary onset activations, both of which are instantaneous attributes. To consider the metrical structure of music, this model is trained from aligned paired data annotated with tatum, beat, and downbeat times such that the pitches, onset activations, and beat and downbeat activations are jointly predicted at the tatum level. We experimentally report the performance and remaining problems of the proposed method.

Chapter 7 concludes this thesis with a brief look at future work. Further investigation is needed to address musical notes with irregular rhythms (*e.g.*, triples), time signature changes, and multiple vocal parts.

Acknowledgments

This work was accomplished at Speech and Audio Processing Lab., Graduate School of Informatics, Kyoto University. I express my gratitude to all people who helped me and this work.

At first, I would like to express my special thanks and appreciation to my supervisor Associate Professor Kazuyoshi Yoshii. His comments were essential and insightful for advancing this work. This work would not have been completed without his continuing engagement and generous support.

I also express my special thanks and appreciation to Professor Tatsuya Kawahara. He gave me a lot of essential and insightful comments on my research in our laboratory meetings.

Furthermore, I express my special thanks and appreciation to the members of my dissertation committee, Professor Ko Nishino and Professor Hisashi Kashima for their time and valuable comments and suggestions.

This thesis cannot be accomplished without continuing engagement and generous support of Assistant Professor Eita Nakamura. He gave me insightful advice from his deep knowledge of machine learning and mathematics. He also gave much time to meaningful discussions.

I would like to thank Dr. Masataka Goto, Dr. Tomoyasu Nakano, and Dr. Satoru Fukayama who are the members of Media Interaction Group, Human Informatics and Interaction Research Institute (HIIRI), National Institute of Advanced Industrial Science and Technology (AIST).

I also deeply thank both current and past members in Speech and Audio Processing Lab. I am grateful for comments and supports from Specially Appointed Associate Professor Katsutoshi Itoyama, Assistant Professor Koji Inoue,

Dr. Yoshiaki Bando, Dr. Kohei Sekiguchi, Mr. Wu Yiming, Mr. Hirofumi Inaguma, Mr. Sei Ueno, the members of the music group, and the other members.

This work was supported by the Japan Society for the Promotion and Science (JSPS) with their financial support as a Fellowship for Young Scientists (DC2).

Last but not least, I am truly grateful to my family for their support of my long student life.

Contents

Abstract i

Acknowledgments iii

Contents viii

List of Figures xiii

List of Tables xv

1 Introduction 1

1.1 Background 1

1.2 Music Representations 2

1.2.1 Musical Score Representation 2

1.2.2 Piano-Roll Representation 4

1.3 Requirements 4

1.4 Approaches 5

1.4.1 Generative Approach Based on HSMM 6

1.4.2 Hybrid Approach Based on CRNN-HSMM 7

1.4.3 Discriminative Approach Based on Encoder-Decoder Model with Note-Level Output 7

1.4.4 Discriminative Approach Based on Encoder-Decoder Model with Tatum-Level Output 8

1.4.5 Organization 9

2 Literature Review **11**
2.1 Automatic Music Transcription 11
2.1.1 Piano-Roll Estimation 11
2.1.2 Musical Score Estimation 12
2.2 Sequence-to-Sequence Learning 13
2.2.1 Generative and Hybrid Approaches 14
2.2.2 Discriminative Approaches 15

3 Generative Approach Based on HSMM **19**
3.1 Introduction 19
3.2 Method 22
3.2.1 Problem Specification 23
3.2.2 Musical Score Model 25
3.2.3 F0 Trajectory Model 27
3.2.4 Bayesian Formulation 29
3.3 Training and Inference 31
3.3.1 Unsupervised Learning 31
3.3.2 Semi-supervised Learning 35
3.3.3 Posterior Maximization 36
3.4 Evaluation 38
3.4.1 Experimental Conditions 38
3.4.2 Experimental Results 42
3.4.3 Further Investigations 49
3.5 Summary 51

4 Hybrid Approach Based on CRNN-HSMM **53**
4.1 Introduction 53
4.2 Method 55
4.2.1 Problem Specification 56
4.2.2 Generative Modeling Approach 57
4.2.3 Language Model 58
4.2.4 Tatum-Level Language Model Formulation 60

4.2.5 Acoustic Model 62
4.2.6 Training Model Parameters 65
4.2.7 Transcription Algorithm 65
4.3 Evaluation 67
4.3.1 Data 67
4.3.2 Setup 68
4.3.3 Method Comparison 70
4.3.4 Influences of Voice Separation and Beat Tracking Methods 72
4.3.5 Discussion 73
4.4 Summary 75

5 Discriminative Approach Based on Encoder-Decoder Model with Note-Level Output 77
5.1 Introduction 77
5.2 Method 79
5.2.1 Problem Specification 79
5.2.2 Pitch and Note Value Decoder 80
5.2.3 Loss Function for Attention Weights 80
5.2.4 Training and Inference Algorithms 81
5.3 Evaluation 81
5.3.1 Experimental Data 82
5.3.2 Configurations 82
5.3.3 Evaluation Metrics 84
5.3.4 Experimental Results 84
5.4 Summary 86

6 Discriminative Approach Based on Encoder-Decoder Model with Tatum-Level Output 89
6.1 Introduction 89
6.2 Method 91
6.2.1 Problem Specification 92
6.2.2 Frame-level Encoders 92

6.2.3 Tatum-level Decoder with an Attention Mechanism 92
6.2.4 Loss Functions for Attention Weights 95
6.3 Evaluation 95
6.3.1 Data 95
6.3.2 Setup 96
6.3.3 Metrics 97
6.3.4 Results 98
6.4 Summary 99

7 Conclusion **101**
7.1 Contributions 101
7.2 Future Work 103

Bibliography **105**

List of Publications **117**

List of Figures

1.1 Symbols and terms used in a musical score representation. 3

1.2 Organization of this thesis. 10

3.1 The problem of automatic singing transcription. The proposed method takes as input a vocal F0 trajectory and tatum times, and estimates a sequence of musical notes by quantizing the F0 trajectory in the time and frequency directions. 20

3.2 The generative process of a vocal F0 trajectory based on the proposed model consisting of a musical score model and an F0 trajectory model. The musical score model represents the generative process of musical notes (pitches and onset score times) based on local keys assigned to measures. In the figure of musical notes, the black vertical lines represent a tatum grid given as input. The F0 trajectory model represents the generative process of a vocal F0 trajectory from the musical notes by adding the frequency and temporal deviations. In the figure of temporally deviated notes, the arrows represent temporal deviations of onset times from tatum times. 22

3.3 Relationships between different time indices j, n, and t. The upper figure shows the start and end beat times of each musical note indexed by j. The dotted lines between the upper and lower figures represent correspondence between the tatum index n and the frame index t. The lower figure shows the F0 value of each time frame. The onset of the first note z_1 is the start of music and z_0 is a supplementary note that is used only for calculating the slanted line representing the transient segment of z_1. 23

3.4 A musical score model that represents the generative process of local keys, note pitches, and note onsets. The top row represents the Markov chain of local keys. The second row represents the Markov chain of the note pitches. The third row represents a sequence of musical notes. The bottom represents the Markov chain of the onset score times of musical notes. The vertical lines represent tatum times and the bold ones represent bar lines. . . . 26

3.5 Temporal and frequency deviations in vocal F0 trajectories. In both figures, the black vertical lines represent tatum times. In Fig. (a), the blue and green vertical lines represent the start and end frames of the transient segment of a vocal F0 trajectory. In Fig. (b), the arrows represent the frequency deviations of a vocal F0 trajectory from the frequency of a musical note. 28

3.6 Temporally deviated pitch trajectory used as Cauchy location parameters. The blue and green vertical boxes represent the start and end frames of the transient segment and the grey vertical boxes represent the tatum times of note onsets. The red boxes represent the temporally deviated pitch trajectory of note j and the grey boxes represent the temporally deviated pitch trajectory of the other notes. 30

3.7 A relationship between the variables $q_j = \{p_j, o_j, e_j, d_j\}$ and $\bar{q}_n = \{\bar{p}_n, \bar{o}_n, \bar{e}_n, \bar{d}_n\}$. 33

3.8 Pretrained transition probabilities between the 16th-note-level tatum positions. 42
3.9 Pretrained transition probabilities between the 12 pitch classes under the major and minor diatonic scales. 45
3.10 Transition probabilities between the 12 pitch classes estimated by the unsupervised learning method (Section 3.3.1). 46
3.11 Transition probabilities between the 12 pitch classes estimated by the posterior maximization method (Section 3.3.3). 47
3.12 Estimation errors caused by using the pretrained initial and transition probabilities of pitch classes. The pale blue backgrounds indicate the diatonic scales of estimated keys and the gray boxes indicate ground-truth musical notes. The blue and red lines indicate vocal F0s and estimated musical notes, respectively. The orange dots indicate estimated note onsets. The gray grids indicate tatum times and semitone-level pitches. The red balloons indicate the ground-truth notes that the proposed method failed to estimate. The estimated keys are illustrated in the figure, and the ground-truth key in both examples is D minor. 48
3.13 Positive effects of temporal deviation modeling (cf. Fig. 3.12). The green lines estimated F0s with temporal deviations. The red arrows indicate estimation errors and the green arrows and balloons indicate correct notes obtained by modeling temporal deviations. 49
3.14 Negative effects of temporal deviation modeling (cf. Fig. 3.13). The red balloons indicate estimation errors. 50
3.15 The categorical distribution of the onset time deviations $\mathbf{E}$ and that of the transient durations $\mathbf{D}$ estimated in the unsupervised learning method (Section 3.3.1). 50
3.16 Positive and negative effects of duration penalization (cf. Fig. 3.12). The green and red balloons indicate improved and deteriorated parts. 51

4.1 The problem of automatic singing transcription. The proposed method takes as input a spectrogram of a target music signal and tatum times and estimates a musical score of a sung melody. . . . 54

4.2 The proposed hierarchical probabilistic model that consists of a SMM-based language model representing the generative process of musical notes from local keys and a CRNN-based acoustic model representing the generative process of an observed spectrogram from the musical notes. We aim to infer the latent notes and keys from the observed spectrogram. 57

4.3 Representation of a melody note sequence and variables of the language model. 60

4.4 The acoustic model $p(\mathbf{X}|\overline{\mathbf{P}}, \overline{\mathbf{C}})$ representing the generative process of the spectrogram $\mathbf{X}$ from note pitches $\overline{\mathbf{P}}$ and residual durations $\overline{\mathbf{C}}$. 64

4.5 Architecture of the CNN. Three numbers in the parentheses in each layer indicate the channel size, height, and width of the kernel. 69

4.6 Examples of musical scores estimated by the proposed method, the CRNN method, the HSMM-based method, and the majority-vote method from the mixture and separated audio signals and the estimated F0 contours and tatum times. Transcription errors are indicated by the red squares. Capital letters attached to the red squares represent the following error types: pitch error (P), rhythm error (R), deletion error (D), and insertion error (I). Error labels are not shown in the transcription result by the majority-vote method, which contains too many errors. 71

5.1 Our encoder-decoder model with an attention mechanism for end-to-end AST. This model is trained by minimizing the weighted sum of loss functions for ground-truth pitches and note values, as well as alignment information (onset times) if available. 78

5.2 NERs calculated on the validation data during the training. Grey lines indicate NERs of each iteration, and colored lines indicate the average values of the NERs for the past 100 iterations. 85
5.3 WERs with different usage rates of training data $\mathbf{Z}$. 85
5.4 Examples of attention weights and musical notes estimated by the proposed method. Red, blue, yellow, and green horizontal lines indicate musical notes, grey lines indicate rests, and black squares indicate the onset positions of the musical notes. The top two figures are the input spectrogram and the ground-truth musical notes. The subsequent figures are attention weights and musical notes for $\lambda = 1$, $\lambda = 0$, and the gradual reduction of λ from top to bottom. 87

6.1 The proposed neural encoder-decoder model with a beat-synchronous attention mechanism for end-to-end singing transcription. DB = 'downbeat'. New loss functions for the centroids of attention weights are introduced to align them with equally-spanned beat times. 90

List of Tables

3.1 Performance of the proposed method with different learning and model configurations. 43
3.2 Performance of the conventional and proposed methods. 44
3.3 Performance of the proposed method based on E0 estimation and/or tatum detection. 44

4.1 The AST performances (%) of the different methods. 70
4.2 The AST performances (%) obtained from the different input data. 72

5.1 Word error rates on the test data. 83
5.2 Note-level error rates on the test data. 84

6.1 Error rates [%] in tatum and note levels. 96

Chapter 1

Introduction

This chapter describes the problem of audio-to-score singing transcription for music signals and explains our approaches.

1.1 Background

Transcribing music is essential to investigate the mechanism of human intelligence for sound recognition. Music is a considerable complex signal that has multiple overlapping sound elements with structure in frequency and temporal directions. Humans can recognize individual sound elements in music and describe them in symbolic forms (*i.e.*, music notations). However, the realization of the music recognition mechanism is challenging because the design of computational algorithms to convert music signals into music notations comprises several subtasks like the separation of instrument parts, pitch and timing detection of each sound element, and beat and rhythm tracking [1, 2].

Automatic music transcription (AMT) is one of the most fundamental recognition tasks in the field of music information processing. The ultimate goal of AMT is to estimate human-readable and playable musical scores consisting of multiple musical instrument parts from music signals. The musical score is the most common format for describing, archiving, and distributing a wide variety of Western tonal music including popular music, which is focused on in this thesis. If one wants to play his or her favorite popular songs, it would be necessary to buy manually-transcribed musical scores (band scores) at a bookstore.

However, such scores are provided for only a limited amount of commercial songs. An alternative way is to manually transcribe musical scores, but it is very hard or time-consuming even for musically-trained people.

This thesis addresses automatic singing transcription (AST) that aims to estimate a musical score of the sung melody from an audio signal of popular music. The singing voice plays an important role in popular music because it usually forms the melody line and influences the impression of the song. Many studies have been conducted for recognition and generation of singing voice such as melody extraction (F0 trajectory estimation) [3–12], singing voice separation [8, 11, 13–18], and singing voice synthesis [19]. Transcribed musical scores can be used for various applications such as query-by-humming, music retrieval, musical grammar analysis [20], score-informed singing separation, singing voice synthesis, and active music listening [21].

1.2 Music Representations

We explain two major representations used for describing music in a symbolic format; a musical score (sheet music) representation and a piano-roll representation [22, 23]. We also introduce music-specific symbols and terms.

1.2.1 Musical Score Representation

The musical score is a human-friendly representation of music. Whereas the *pitch* is usually considered as a perceptual attribute about the ordering of sounds on a frequency-related scale (1.1-(a)), in this thesis, it is defined as a physical attribute consisting of a *pitch class* and an *octave*, where the interval between consecutive pitches is called a *semitone*, the octave represents an interval consisting of twelve semitones, and the pitch class is one of the twelve different pitches $\{\mathrm{C}, \mathrm{C}\sharp/\mathrm{D}\flat, \ldots, \mathrm{B}\}$ in one octave. The accidental notations such as *sharp* ($\sharp$) and *flat* ($\flat$) are used for raising and lowering a pitch by a semitone, respectively.

Each musical note is put on the five horizontal lines called *staff* with a *clef* (1.1-(b)), where the clef is described at the start of the staff and the center of the

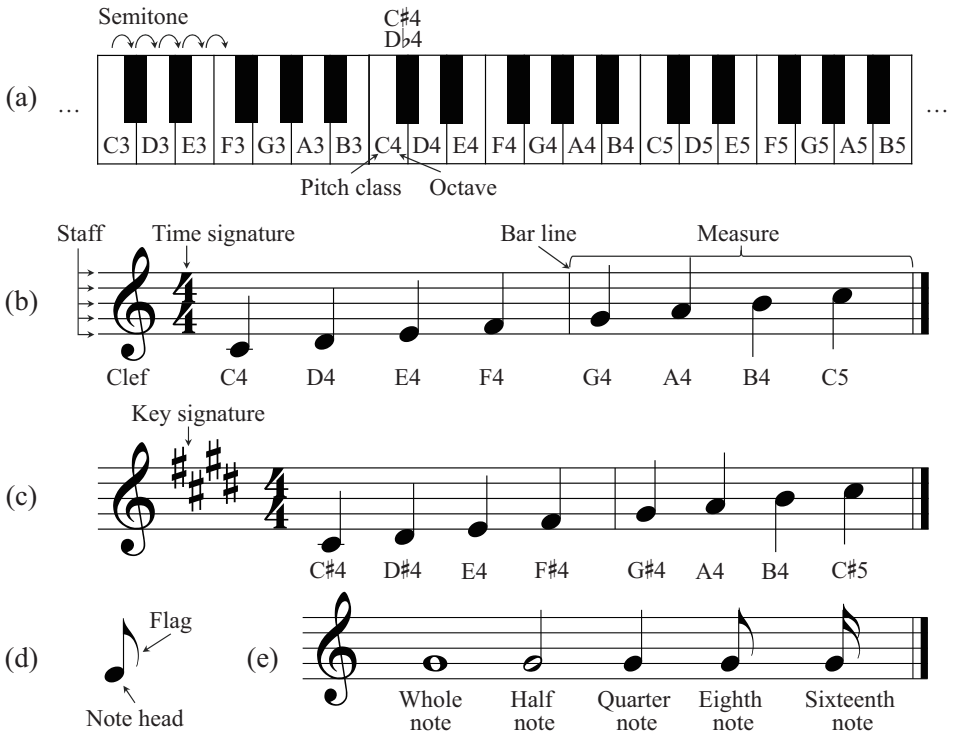


Figure 1.1: Symbols and terms used in a musical score representation.

clef indicates the pitch of G4. A *key signature* representing global pitch shifting is often placed after the clef (1.1-(c)). The key signature is described as a set of either sharps or flats.

A *note value* is the duration of a musical note represented by the color of a *note head* and the number of *flags* (1.1-(d)). Given that a whole note has a reference length (1.1-(e)), a half note has half the length of the whole note, a quarter note has quarter the length of the whole note, and so on. Similarly, the *rest* family (*e.g.*, whole, half, and quarter rests) that represents the silence of a specified duration is defined.

The metrical structure of a musical score is described by *downbeat*s and *meter*s. Each downbeat position is represented as a vertical line called a *bar line* on a musical score, and the segment between consecutive bar lines is called a *measure* (1.1-(b)). The length of each measure is determined by the meter described by a *time signature*. The numerator and denominator of the time signature represent a note value corresponding to one beat and the number of beats in one measure,

respectively. The term *tatum* used in this thesis is determined as the minimum unit to represent note values.

1.2.2 Piano-Roll Representation

The piano roll is a computer-friendly representation of music. It is represented as a two-dimensional matrix whose vertical and horizontal axes represent quantized semitone-level pitches and times in seconds, respectively. Each note event is represented as a tuple of a semitone-level pitch, an onset time, and an offset time (or duration) on the two-dimensional matrix. The piano-roll representation is the basis of the standard MIDI format, which can be synchronized with the audio spectrogram. In conventional studies, techniques proposed in the field of image processing have often been for converting an audio spectrogram into a piano roll, both of which are represented at the frame level.

1.3 Requirements

We discuss three fundamental requirements that should be taken into account for developing an audio-to-score AST system.

Representation of Singing Deviations: The singing voice has a continuous F0 trajectory that significantly deviates from F0s and temporal positions specified by musical notes on a musical score. From a physical point of view, the F0 trajectory is fated to smoothly change from one note to another and slightly fluctuate at the middle part of a musical note because of the continuous movement of a throat over time. In addition, the F0s are often modulated actively according to singing expressions such as portamento and glissando (F0 sliding from one note to another) and vibrato (regular and pulsating F0 change). When a singer intentionally sings a song ahead of the beat or behind the beat, the actual note onsets, which are perceived subjectively, would be shifted forward or backward as a whole. Therefore, the naive quantization of the F0 trajectory on the regular time-frequency grids yields many error notes. This calls for deviation-robust representation of singing voice.

Representation of Grammatical Structures: Sequences of musical notes in Western tonal music have grammatical structure consistent with music theory. For example, consecutive notes have sequential dependency and the relative frequencies of the 12 pitch classes are affected by the underlying keys. The musical rhythm emerges from a sequence of note values and the rhythmic patterns of notes are characterized by the relative positions of note onsets in metrical structure (*i.e.*, beats and downbeats). We use the grammatical structure as a clue for inferring musically-natural note sequences from music, just as we use linguistic knowledge to recognize word sequences from speech. If only the acoustic features of singing voices are focused on, a number of out-of-scale pitches and irregular rhythms are included in the estimated note sequence because of the frequency and temporal deviations of singing voice. This calls for learning the grammatical structure of musical notes.

Direct Estimation of Musical Notes: Considering the remarkable progress of vocal F0 estimation (melody extraction) [3–12] and beat tracking [24], one might first estimate a vocal F0 trajectory and beat and downbeat times from a music signal and then estimate a note sequence by quantizing the F0 trajectory. Such a cascading approach, however, has two major problems. First, the acoustic features of the original singing voice (*e.g.*, volumes, spectral envelopes, and harmonic structures) cannot be used for the note estimation. This makes it difficult to recognize consecutive musical notes with the same pitch. Second, the errors of the F0 estimation and beat tracking adversely affect the subsequent note estimation. This calls for direct estimation of musical notes from music signals bypassing F0 estimation and beat tracking.

1.4 Approaches

In this thesis, we take a principled approach to audio-to-score AST based on integration of a language model describing the grammatical structure of notes and an acoustic model describing the deviations of singing voice from notes. Because each model can be formulated in either a generative or discriminative

manner, we propose generative, hybrid, and discriminative unified models, each of which consists of language and acoustic models. The key feature common to these unified models is that the most likely note sequence is estimated from a vocal F0 trajectory or spectrogram at once, while considering both the musical naturalness of notes and their fitness to the trajectory or spectrogram.

1.4.1 Generative Approach Based on HSMM

In Chapter 3, we first investigate the effectiveness of the language model representing the *grammatical structures* and the acoustic model representing the *singing deviation* for estimating musical notes from a vocal F0 trajectory with tatum times. The melody extraction (i.e., the estimation of a vocal F0 trajectory) and beat tracking (i.e., the estimation of tatum times) have been well studied, and the vocal F0 trajectory and tatum times provide enough information on pitches and durations of musical notes. A straightforward approach is to determine the note pitches by taking the majority of quantized F0s in each tatum interval. This approach, however, has no mechanism that avoids out-of-scale pitches and irregular rhythms caused by the considerable singing deviations.

To realize AST robust against the singing deviations, we take a generative approach similar to the statistical speech recognition approach based on a *language model* and an *acoustic model*. We formulate a hierarchical hidden semi-Markov model (HSMM) of a vocal F0 trajectory that consists of a *generative* language model describing the transitions of local keys and the rhythms of notes and a *generative* semi-acoustic model precisely describing the time-frequency singing deviations of the trajectory. Given an F0 trajectory and tatum times with metrical structure (*i.e.*, meters and downbeats), a sequence of musical notes, that of local keys, and the temporal and frequency deviations can be estimated jointly by using a Markov chain Monte Carlo (MCMC) method while leveraging both the grammatical knowledge described by the language model and the singing deviations described by the acoustic model. Thanks to the language model evaluating the grammatical structure of the note sequence, musically-unnatural notes can be avoided effectively.

1.4.2 Hybrid Approach Based on CRNN-HSMM

In Chapter 4, we investigate the effectiveness of the acoustic model for the *direct estimation of musical notes* in addition to the language model representing the *grammatical structures*. The generative approach in Chapter 3 described the singing deviation in a vocal F0 trajectory by the acoustic model and improved the AST performances by leveraging the grammatical knowledge represented the language model. In the generative approach, however, the F0 estimation errors propagate to the note estimation step, and rich acoustic information cannot be used. For example, repeated notes of the same pitch cannot be detected from only F0 information because the F0 information cannot includes onset information of musical notes. To investigate the method that can avoid the error propagation and utilize the full information of singing voices, it is necessary to construct the acoustic model that can directly handle music spectrograms.

We then formulate an HSMM of a vocal spectrogram that consists of a *generative* language model similar to that in Chapter 3 and a *discriminative* acoustic model based on a convolutional recurrent neural network (CRNN) trained in a supervised manner for predicting the posterior probabilities of note pitches and onsets from the spectrogram. Musical notes including consecutive notes of the same pitch and rests can be directly estimated without using F0 estimation. Given a vocal spectrogram and tatum times with metrical structure (*i.e.*, meters and downbeats), the most-likely note sequence is estimated with the Viterbi algorithm, while leveraging both the grammatical knowledge about musical notes and the expressive power of the CRNN. The proposed CRNN-HSMM can achieve the state-of-the-art performance thanks to the combination of the key- and rhythm-aware regularization of the estimated note sequence and the robustness of the CRNN against the large variations of singing voices.

1.4.3 Discriminative Approach Based on Encoder-Decoder Model with Note-Level Output

In Chapter 5, we investigate the integration of a *discriminative* acoustic model and a *discriminative* language model based on sequence-to-sequence learning for

musical note estimation. The generative and hybrid approaches to AST require vocal F0 trajectories or tatum times estimated in advance. In these approaches, however, the F0 and tatum estimation errors propagate to the note estimation step. In addition, it is non-trivial to split continuous singing voice into segments corresponding to musical notes for making precise time-aligned transcriptions. We attempt to use the standard encoder-decoder architecture with the attention mechanism consisting of a frame-level encoder and a *note-level* decoder, where the encoder and decoder are considered to work as *discriminative* acoustic and language models, respectively. The encoder-decoder model can be trained from non-aligned data without referring to tatum times and *directly estimate musical notes* from music spectrograms.

The main challenge of this study is to estimate temporal attributes (note values), which are not handled in ASR, in addition to instantaneous attributes (pitches) in the attention-based encoder-decoder framework. In a preliminary experiment, we found that the encoder-decoder model has weakness in predicting temporal attributes and that the accurate estimation of attention weights is crucial. To solve this problem, we also propose a semi-supervised learning framework based on a loss function for an attention matrix that encourages each note in an output sequence to attend the onset frame in an input sequence. This framework is inspired by the existing method [25–28] imposing structural constraints on the attention weights and effectively uses fewer aligned paired data for learning monotonic alignment between input and output sequences. Furthermore, we also introduce weakly-supervised learning that gradually reduces the weight of the attention loss for better input-output alignment. We experimentally show the effectiveness of the semi- and weakly-supervised frameworks for improving AST performances.

1.4.4 Discriminative Approach Based on Encoder-Decoder Model with Tatum-Level Output

In Chapter 6, we also investigate the integration of a *discriminative* acoustic model and a *discriminative* language model based on sequence-to-sequence learning for

AST. A promising approach to such sequence-to-sequence learning is to use an encoder-decoder model with an attention mechanism. This approach, however, cannot be used straightforwardly for singing transcription because a *note-level* decoder fails to estimate note values from latent representations obtained by a *frame-level* encoder that is good at extracting instantaneous features, but poor at extracting temporal features. To solve this problem, in Chapter 5, we proposed the semi-supervised learning framework that imposes the alignment constraints on the attention matrix by using a limited amount of fewer aligned paired data. However, it is time-consuming to make precise time-aligned pair data of music signals and musical notes. In addition, the discriminative approach in Chapter 5 does not predict the metrical structure (*i.e.*, meters and downbeats) required for reconstructing complete musical scores.

To solve this problem, we propose a new encoder-decoder model consisting of a frame-level encoder and a *tatum-level* decoder for directly estimating sequences of pitches and binary onset activations, both of which are instantaneous attributes. To consider the metrical structure of music, this model is trained from aligned data annotated with tatum, beat, and downbeat times such that the pitches, onset activations, and beat and downbeat activations are jointly predicted at the tatum level. In addition, to investigate the guiding mechanism of an attention matrix without using time-aligned data of music signals and musical notes, we propose a beat-synchronous attention mechanism for monotonically aligning tatum-level scores with input audio signals with a steady increment. We experimentally report the performance and remaining problems of the proposed method.

1.4.5 Organization

The organization of this thesis is outlined in Fig. 1.2. Chapter 2 reviews related work on estimating piano-roll and musical score representations from music signals. Chapter 3 presents a hierarchical hidden semi-Markov model (HSMM) to estimate a musical score from a vocal F0 trajectory under the condition that tatum times are given in advance. Chapter 4 presents a hybrid model of a

	Chapter 3	**Chapter 4**	**Chapter 5**	**Chapter 6**
Model name	HSMM	CRNN-HSMM hybrid model	Attention-based encoder-decoder model	
Language model	Semi-Markov model (**Generative**)	Semi-Markov model (**Generative**)	Long short-term memory (**Discriminative**)	
Acoustic model	Cauchy and discrete distributions (**Generative**)	Convolutional recurrent neural network (**Discriminative**)	Long short-term memory (**Discriminative**)	
Input signal	F0 trajectory	Spectrogram		
Tatum times	Given		Not given	
Metrical structure	Given		Not estimated	Estimated

Figure 1.2: Organization of this thesis.

deep neural network and a hidden Markov model to estimate a musical score directly from a music signal under the condition that tatum times are given in advance. Chapter 5 presents an attention-based encoder-decoder model for an end-to-end melody note estimation. Chapter 6 presents another attention-based encoder-decoder model to jointly estimate musical notes and metrical structure. Chapter 7 concludes this thesis with future directions.

Chapter 2

Literature Review

This chapter reviews related work on automatic music transcription.

2.1 Automatic Music Transcription

This section first introduces existing methods for estimating the piano-roll representations and the musical scores from music audio signals.

2.1.1 Piano-Roll Estimation

Many studies have attempted to convert music audio signals into the piano-roll representations. In the piano-roll representation, only pitches of musical notes are quantized at the semitone level, and onset times and durations of musical notes are represented at the frame or second level. The piano-roll estimation for singing voice is usually performed for F0 trajectories estimated in advance and includes two sub-tasks: the detection of note segments (onset and offset times) and the estimation of quantized pitches in the note segments.

Some studies have estimated the piano-roll representation of singing voices based on hand-crafted rules and filters [29,30]. Hidden Markov models (HMMs) are used for jointly conducting the note segment detection and the quantized pitch estimation. Ryynänen *et al.* [31] proposed a method based on a hierarchical HMM that represents the generative process of an F0 trajectory. In this model, the upper-level HMM represents the transition between quantized pitches, and the lower-level HMM represents the transition between attack, sustain, and release

states of each note. Mauch *et al.* [32] developed a software tool called Tony for analyzing extracting pitches. This tool extracts a vocal F0 trajectory by pYIN [7] and estimates musical notes by a modified version of Ryynänen's method [31]. Yang *et al.* [33] also proposed a method based on a hierarchical HMM with the three internal states of each note that represents the generative process of f0-Δf0 planes. A DNN-based method for the note segment detection was reported [34]. This method estimates the quantized pitches by taking medians of F0s in the individual note segments.

The piano-roll estimation directly from input spectrograms has been conducted for polyphonic music signals such as piano performance and vocal quartet. Spectrogram factorization techniques like probabilistic latent component analysis (PLCA) and non-negative matrix factorization (NMF) are used for estimate discrete pitches of each time frame for a piano [35–37] and a vocal quartet [38], followed by note tracking based on HMMs. DNNs [39, 40] have recently been employed for estimating multiple discrete pitches of each time frame. Other DNN-based methods [41, 42] jointly estimated pitch and onset activations of each time frame to obtain note events.

2.1.2 Musical Score Estimation

There are several approaches to estimating musical scores, where each musical note is described as a tuple of a pitch quantized in semitones and a note value quantized in musical units (*i.e.*, tatums). One of the typical approaches to this problem is rhythm transcription, which takes a piano-roll representation as input and estimates note values by removing the temporal deviations of onset and offset times of each note event in the piano-roll representation. Several approaches for the rhythm transcription have been studied based on hand-crafted rules [43], a connectionist model [44], probabilistic context-free grammars [45], and hidden Markov models [46–50].

The HMM-based approaches are categorized into two types: a duration-based HMM and an onset-based HMM. The duration-based HMM [46] represents note values and local tempos as latent variables and note durations as

observed variables. The observed duration is described as a product of the note value and the tempo. The onset-based HMM [47–50], which is called a metrical HMM, represents note onset position on beat (tatum) grid as latent variables and onset times as observed variables. The note values are obtained as differences between successive note onset positions on the beat grids. In addition, the metrical HMM has the advantage of estimating the meter and bar lines and avoiding grammatically incorrect score representations (e.g., incomplete triplet notes). To achieve a complete audio-to-score system that estimates a musical score from polyphonic piano signals, the cascading methods that combine the piano-roll estimation and the rhythm transcription have been proposed [51, 52].

Inspired by the recent remarkable progress of deep neural networks, end-to-end approaches that estimate musical symbols directly from input audio signals have emerged. Carvalho *et al.* [53] proposed a method based on the sequence-to-sequence model [54] that predicts the symbols of Lilypond format [55] from features extracted from an audio signal of a synthesized piano sound by using one-dimensional CNNs. Roṁan *et al.* [56] proposed a monophonic transcription method based on the connectionist temporal classification (CTC) that predicts the symbols of Plaine & Easie Code (PAE) format from the magnitude spectrograms of synthesized piano sounds. They also proposed a polyphonic transcription method [57] based on the CTC that predicts **Kern-format-based symbols from a dataset of four-voice chorales synthesized with pipe organ sounds and a dataset of four-voice quartets synthesized with string sounds. In addition, they investigated a CTC-friendly format by compared several symbolic formats for describing music [58].

2.2 Sequence-to-Sequence Learning

This section introduces approaches for converting a sequence into another sequence while focusing mainly on automatic speech recognition (ASR), which is similar to AST in that they map a sequence of acoustic features into a sequence of discrete symbols.

2.2.1 Generative and Hybrid Approaches

Let $\mathbf{X} = [x_1, \ldots, x_T]$ and $\mathbf{Y} = [y_1, \ldots, y_N]$ be input and output sequences, where T and N are their lengths, respectively. In the probabilistic framework, the output sequence $\mathbf{Y}$ is inferred from the input sequence $\mathbf{X}$ as a maximum a posteriori probability (MAP) estimation as follows:

$$\mathbf{Y} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, p(\mathbf{Y}|\mathbf{X}). \tag{2.1}$$

In addition, by using the Bayes' theorem defined as

$$p(\mathbf{Y}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y})}{p(\mathbf{X})} \propto p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y}). \tag{2.2}$$

Eq. (2.1) is formulated as follows:

$$\mathbf{Y} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y}). \tag{2.3}$$

In an ASR system based on the MAP approach, $\mathbf{X}$ and $\mathbf{Y}$ are set to an acoustic feature sequence and a word sequence $\mathbf{W}$, and $p(\mathbf{X}|\mathbf{W})$ and $p(\mathbf{W})$ are called an *acoustic model* and a *language model*, which represent the generative process of the acoustic features given the words and the generative process of the word sequence, respectively. Furthermore, the state-of-the-art ASR system is based on the DNN-HMM hybrid model that trains the acoustic model using deep neural networks [59–63].

This MAP approach has been used in the field of music information processing. For example, vocal F0 estimation is performed by setting $\mathbf{X}$ and $\mathbf{Y}$ to music acoustic features and F0 trajectories [5, 64]. In Chapter 3, by letting $\mathbf{X}$ and $\mathbf{Y}$ be vocal F0 trajectory and musical notes, we integrate the music *language* model describing the generative process of musical notes given local keys with the *acoustic* model describing the time-frequency fluctuations of a vocal F0 trajectory from musical notes. In Chapter 4, inspired by the DNN-HMM hybrid model in AST, we also propose the *discriminative acoustic* model based on a convolutional recurrent neural network (CRNN) trained in a supervised manner for predicting the posterior probabilities of note pitches and onsets from the input music spectrogram.

2.2.2 Discriminative Approaches

The end-to-end approach to converting an input sequence into an output sequence has emerged in the field of machine translation [54, 65]. This approach is typically composed of two recurrent neural networks (RNNs) called an encoder and a decoder. The RNN encoder summarizes the input sequence into a single vector representation. The RNN decoder recursively predicts the output sequence from the vector representation. The advantage of this approach lies in its simple architecture for optimizing the whole parameters at once from a variable input-output sequence. However, the single vector is not expressive enough to represent the information of an entire long sequence.

An attention-based encoder-decoder model has recently been proposed for machine translation [66, 67] and automatic speech recognition [68–71] to overcome the disadvantage of the straight encoder-decoder model. The attention-based encoder-decoder model has additional layer computing attention weights, which represent a score matching the hidden state of the RNN decoder to each location of the input sequence. The encoder transforms a sequence of feature vectors (input data) $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{F \times T}$ into a sequence of latent representations $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_T] \in \mathbb{R}^{E \times T}$, where T, F, and E indicate the length of the input sequence, the dimension of the feature vectors, and the dimension of the latent vectors, respectively. The decoder predicts a sequence $\mathbf{Y} = [y_1, \ldots, y_N]$ from the latent vectors $\mathbf{H}$, where N indicates the number of symbols predicted by the decoder. $y_n \in \{1, \ldots, I\}$ indicates the n-th predicted element, where I indicates the vocabulary size of the decoder. The vocabulary includes two special symbols: $\langle \mathsf{sos} \rangle$ and $\langle \mathsf{eos} \rangle$. The attention-based decoder consists of a unidirectional RNN and recursively calculates the following steps:

$$\boldsymbol{\alpha}_n = \mathrm{Attend}(\mathbf{s}_{n-1}, \boldsymbol{\alpha}_{n-1}, \mathbf{H}), \tag{2.4}$$

$$\mathbf{g}_n = \sum_{t=1}^{T} \alpha_{nt} \mathbf{h}_t, \tag{2.5}$$

$$y_n = \mathrm{Generate}(\mathbf{s}_{n-1}, \mathbf{g}_n), \tag{2.6}$$

$$\mathbf{s}_n = \mathrm{Recurrency}(\mathbf{s}_{n-1}, \mathbf{g}_n, y_n), \tag{2.7}$$

where $\mathbf{s}_n \in \mathbb{R}^D$ indicates the n-th hidden state of the decoder, and Attend, Generate, and Recurrency are functions that perform operations on vectors and matrices. Eq. (2.5) represents the attention mechanism. The attention weight $\boldsymbol{\alpha}_n \in \mathbb{R}^T$ is a vector of normalized weights representing the degrees of relevance between the input sequence $\mathbf{X}$ and an output y_n. Each element of $\boldsymbol{\alpha}_n$ is given by

$$\alpha_{nt} = \frac{\exp(e_{nt})}{\sum_{t'=1}^{T} \exp(e_{nt'})}, \tag{2.8}$$

$$e_{nt} = \mathrm{Score}(\mathbf{s}_{n-1}, \mathbf{h}_t, \boldsymbol{\alpha}_{n-1}), \tag{2.9}$$

where Score is a function that calculates a raw weight. In this thesis, we use a convolutional function [69] given by

$$\mathbf{f}_n = \mathbf{F} * \boldsymbol{\alpha}_{n-1}, \tag{2.10}$$

$$e_{nt} = \mathbf{w}^\top \tanh\left(\mathbf{W}\mathbf{s}_{n-1} + \mathbf{V}\mathbf{h}_t + \mathbf{U}\mathbf{f}_{nt} + \mathbf{b}^{\mathrm{Att}}\right), \tag{2.11}$$

where $\mathbf{F} \in \mathbb{R}^{C \times F}$ is a convolutional filter, $\mathbf{f}_n \in \mathbb{R}^{T \times C}$ is the result of the convolution, and C and F indicate the number of channels and the size of the filter. $\mathbf{w} \in \mathbb{R}^A$ indicates a weight vector, $\mathbf{W} \in \mathbb{R}^{A \times D}$, $\mathbf{V} \in \mathbb{R}^{A \times E}$, and $\mathbf{U} \in \mathbb{R}^{A \times C}$ represent weight matrices, and $\mathbf{b}^{\mathrm{Att}} \in \mathbb{R}^A$ represents a bias vector. Here, A is the number of rows of $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{U}$, as well as the number of elements of $\mathbf{b}^{\mathrm{Att}}$. Eq. (2.6) represents the generation of y_n from the previous hidden state $\mathbf{s}_{n-1}$ and the weighted sum $\mathbf{g}_n$ as follows:

$$\boldsymbol{\pi} = \mathrm{Softmax}\left(\mathbf{P}\mathbf{s}_{n-1} + \mathbf{Q}\mathbf{g}_n + \mathbf{b}^{\mathrm{Gen}}\right), \tag{2.12}$$

$$y_n = \underset{y_n \in \{1,\ldots,K\}}{\mathrm{argmax}} \left(\pi_{y_n}\right), \tag{2.13}$$

where $\mathbf{P} \in \mathbb{R}^{I \times D}$, $\mathbf{Q} \in \mathbb{R}^{I \times E}$ represent weight matrices, and $\mathbf{b}^{\mathrm{Gen}} \in \mathbb{R}^I$ is a bias vector. Eq. (2.7) represents the calculation of the next state $\mathbf{s}_n$. Note that the ground-truth symbol is used as y_n in the training phase, whereas in the inference phase, y_n is predicted by the decoder at the previous step and the symbol prediction stops when the output sequence reaches a specified maximum length or when $\langle \texttt{eos} \rangle$ is generated.

The attention weight matrix $\boldsymbol{\alpha} \in \mathbb{R}^{N \times T}$ can be interpreted to represent the alignment between the input and output sequences. For example, for the specific

sequence-to-sequence tasks such as ASR, text to speech (TTS), and AST, the attention weight show that the aligned locations of output elements into the input sequence monotonically line up in ascending order. Therefore, several studies have proposed attention mechanisms that impose constraints on the attention weights to have desirable properties such as monotonicity. Raffel *et al.* [25] and Chui *et al.* [26] proposed the attention mechanism that explicitly enforces a monotonic input-output alignment for online and linear-time decoding for ASR. Tjandra *et al.* [27] proposed an attention mechanism for the monotonicity that computes the difference of the adjacent aligned locations from each hidden state of the RNN decoder. Tachibana *et al.* [28] introduced the guide attention matrix $\mathbf{W} = (w_{nt}) \in \mathbb{R}^{N \times T}$ for TTS, where $w_{nt} = 1 - \exp\{-(n/N - t/T)^2/2g^2\}$. By calculating the loss values between $\mathbf{W}$ and $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ is prompted to become nearly diagonal. Motivated by those methods mentioned above, Chapter 5 proposes the semi- and weakly-supervised learning framework of attention weights using the guide attention matrix calculated from onset times of musical notes obtained in advance. In addition, Chapter 6 proposes another novel attention mechanism for the monotonicity property that minimizes the loss functions calculated from only the attention weights.

Chapter 3

Generative Approach Based on HSMM

This chapter presents the generative approach to AST that estimates a musical score from a vocal F0 trajectory estimated in advance (Fig. 3.1). This approach is based on the hidden semi-Markov model (HSMM) that integrates the *generative* language model with the *generative* acoustic model.

3.1 Introduction

One of the major difficulties of AST is that continuous F0 trajectories include temporal and frequency deviations from straight pitch trajectories indicated in scores. This prohibits a simple quantization method (called *majority-vote method*) that estimates a pitch as the majority of F0s in each tatum interval. A promising way to obtain a natural score is to integrate a *musical score model* (generative language model) that describes the organization of notes in scores with an *F0 trajectory model* (generative acoustic model) representing the temporal and frequency deviations. This framework is similar to the statistical speech recognition approach based on a *language model* and an *acoustic model* [72]. Recent studies have applied musical score models for music transcription in the framework of probabilistic modeling [73,74] and deep learning [75,76].

To build a musical score model, we focus on how pitches and rhythms of musical notes are structured in a sung melody. In tonal music, pitches have

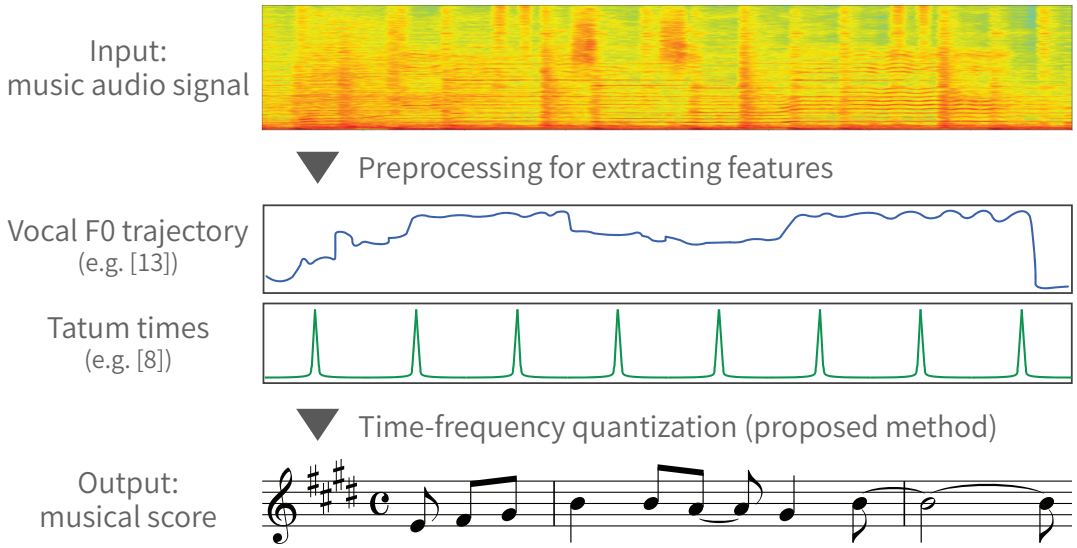


Figure 3.1: The problem of automatic singing transcription. The proposed method takes as input a vocal F0 trajectory and tatum times, and estimates a sequence of musical notes by quantizing the F0 trajectory in the time and frequency directions.

sequential interdependence and are controlled by underlying musical keys or scales. Onset times in scores also have sequential interdependence and are controlled by underlying metrical structure. To represent such characteristics, it is necessary to formulate a musical score model in the musical-note level, instead of in time-frame level [75,76]. On the other hand, a vocal F0 trajectory is represented in the time-frame level, or possibly in the tatum level after applying beat tracking. Because of the mismatch of time scales, integration of a note-level musical score model and a frame- or tatum-level F0/acoustic model poses a challenge in probabilistic modeling, which is still open [77].

For key- and rhythm-aware AST, we previously proposed a hierarchical hidden semi-Markov model (HSMM) [78] that consists of a musical score model and an F0 trajectory model under an condition that the tatum times are given in advance or estimated by a beat detection method [24] (Fig. 3.2). The musical score model generates a note sequence and consists of three sub-models describing local keys, pitches, and onset score times (Section 3.2.2). The local keys are sequentially generated by a Markov model and the pitches of musical notes are

then sequentially generated by another Markov model conditioned on the local keys. The onset score times are sequentially generated by a metrical Markov model [47, 48] defined on the tatum grid. The F0 trajectory model describes the temporal and frequency deviations added to a step-function-like pitch trajectory corresponding to the generated score (Section 3.2.3). To stably learn the musical characteristics unique to each musical piece from a small amount of piece-specific data, the HSMM is formulated in a Bayesian manner (Section 3.2.4).

To estimate a latent sequence of musical notes with decent durations from an observed vocal F0 trajectory by using the HSMM, in this chapter we propose a combination of an iterative Gibbs sampler and a modified Viterbi algorithm that is penalized for intensely favoring longer notes with less frequent transitions (Section 3.3.3). The whole model can be estimated in an unsupervised or semi-supervised manner (Sections 3.3.1 and 3.3.2) by optimizing on the fly or pretraining the musical score model, respectively. Since putting more emphasis on the musical score model was shown to be effective in our previous work [78], in this chapter we carefully optimize the weighting factors on the individual components of the musical score and F0 trajectory models and the note duration penalization with Bayesian optimization [79] or grid search (Section 3.4.1).

The main contributions of this study are as follows. First, we provide a full description of the HSMM (Section 3.2) that is used for transcribing a human-readable score consisting of quantized pitches and onset times from a music audio signal (monophonic F0 trajectory) via the improved learning methods (Section 3.3). This is a principled statistical approach to a well-known open problem of how to integrate a note-level language model with a tatum- or frame-level acoustic model in automatic music transcription. Second, we found that the rhythm and key models of the musical score model and the note duration penalization were particularly effective, by conducting comprehensive comparative experiments for investigating the performances of the unsupervised and semi-supervised learning methods (Section 3.4.2) and evaluating the musical score model (Section 3.4.2), the F0 trajectory model (Section 3.4.2), and the note

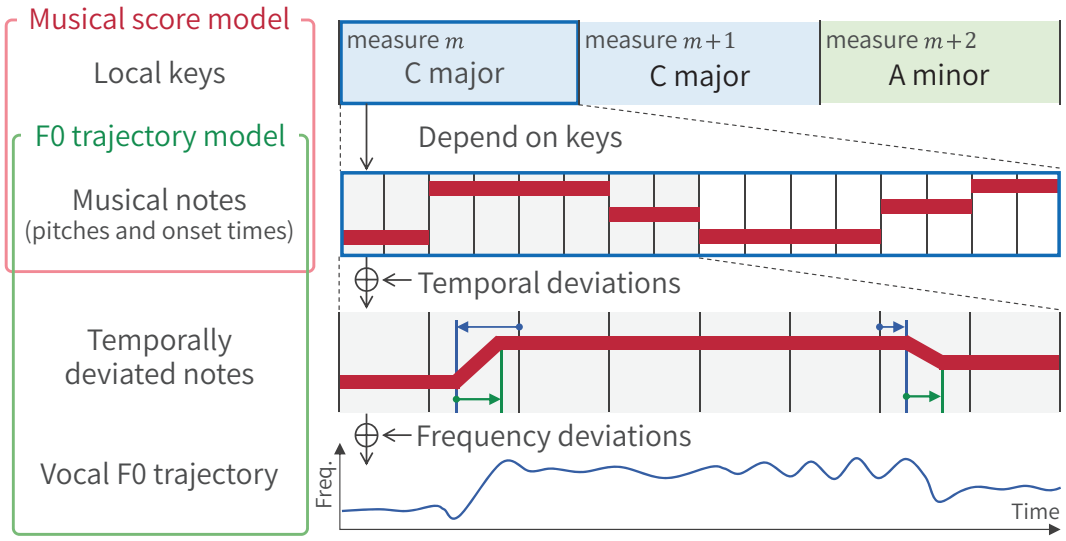


Figure 3.2: The generative process of a vocal F0 trajectory based on the proposed model consisting of a musical score model and an F0 trajectory model. The musical score model represents the generative process of musical notes (pitches and onset score times) based on local keys assigned to measures. In the figure of musical notes, the black vertical lines represent a tatum grid given as input. The F0 trajectory model represents the generative process of a vocal F0 trajectory from the musical notes by adding the frequency and temporal deviations. In the figure of temporally deviated notes, the arrows represent temporal deviations of onset times from tatum times.

duration penalization (Section 3.4.2).

Sections 3.2 and 3.3 describe our statistical approach to AST (generative modeling and posterior inference). Section 3.4 reports the results of comparative experiments. Section 3.5 summarizes this chapter.

3.2 Method

This section defines the task of AST (Section 3.2.1) and explains the hierarchical hidden semi-Markov model (HSMM) that consists of a musical score model and an F0 trajectory model (Fig. 3.2). The musical score model represents the generative process of sung notes in the tatum level (Section 3.2.2) and the F0 trajectory model represents the generative process of vocal F0s in the frame level from the note sequence (Section 3.2.3). We introduce prior distributions to

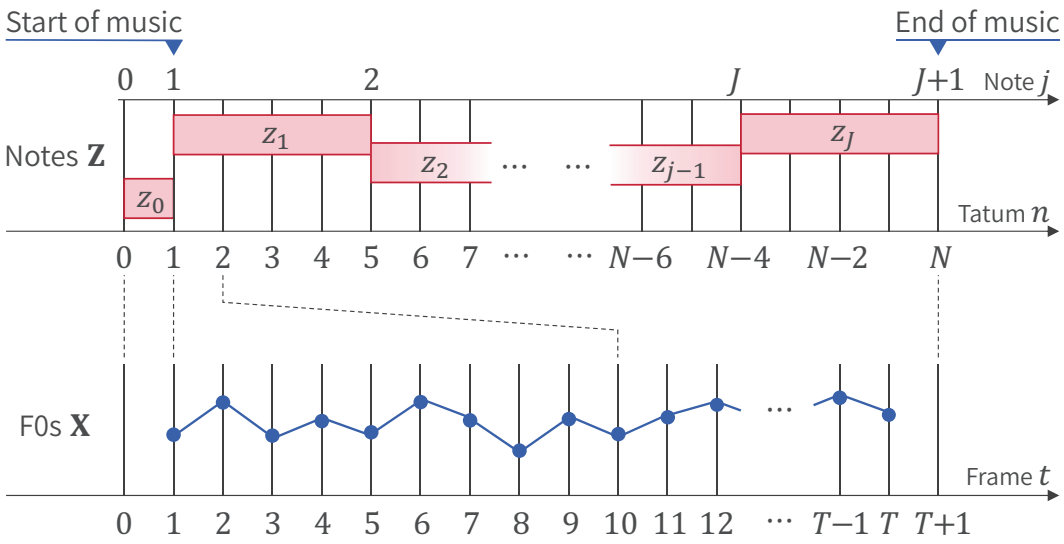


Figure 3.3: Relationships between different time indices j, n, and t. The upper figure shows the start and end beat times of each musical note indexed by j. The dotted lines between the upper and lower figures represent correspondence between the tatum index n and the frame index t. The lower figure shows the F0 value of each time frame. The onset of the first note z_1 is the start of music and z_0 is a supplementary note that is used only for calculating the slanted line representing the transient segment of z_1.

complete Bayesian formulation. This is effective for estimating the reasonable parameters of the proposed model from a small amount of data (Section 3.2.4). We define the meanings of several terms regarding temporal information as follows:

- Onset/offset times and duration: the start/end times and length of a note represented in the frame level.
- Onset/offset score times and note value: the start/end times and length of a note represented in the tatum level.
- Tatum position: the relative position of a tatum in a measure including the tatum.

3.2.1 Problem Specification

Our problem is formalized as follows (Figs. 3.1 and 3.3):

Input:
A frame-level vocal F0 trajectory $\mathbf{X} = x_{0:T}$
and tatum times $\mathbf{Y} = y_{0:N} = (t_n, l_n)_{0:N}$
Output:
A sequence of musical notes $\mathbf{Z} = z_{0:J} = (p_j, o_j)_{0:J}$
By-product:
A sequence of local keys $\mathbf{S} = s_{0:M}$

---

where $x_{0:T} = \{x_0, \ldots, x_T\}$ etc., and T, N, J, and M indicate the number of frames, tatums, estimated notes, and measures, respectively. The time-shifting interval is 10 ms in this study. x_t indicates a log F0 in cents at frame t, where unvoiced frames are represented as $x_t = \mathrm{uv}$. t_n indicates a frame corresponding to tatum n, where $t_0 = 0$, $t_1 = 1$, and $t_N = T + 1$. $l_n \in \{1, \ldots, L\}$ indicates the tatum position, where L is the number of tatums included in a measure ($L = 16$ in this chapter) and $l_n = 1$ indicates the barline. Each note z_j is represented as a pair of a semitone-level pitch $p_j \in \{1, \ldots, K\}$ and an onset score time $o_j \in \{0, \ldots, N\}$, where K is the number of unique pitches considered (*e.g.*, $K = 88$ pitches from A0 to C8), $o_0 = 0$, $o_1 = 1$, $o_{J+1} = N$. We introduce local keys $s_{0:M}$ for each measure. The local key s_m of measure m takes a value in $\{\mathrm{C}, \mathrm{C\#}, \cdots, \mathrm{B}\} \times \{\text{major, minor}\}$ (the tonic is represented as C=0, C#=1, ..., B=11, and the local keys are numbered from 1 to 24). We have introduced supplementary variables x_0, y_0, z_0, and s_0 in order to ease the handling of latent variables at the beginning of music.

In this chapter, we deal with songs in the pop music style. It is assumed that a target piece is in $4/4$ and that tatum unit is 16th note. Rests, notes shorter than the tatum unit, and triplets are not considered. Offset score times are not explicitly modeled, *i.e.*, the offset score time of each note corresponds to the onset score time of the next note. It is also assumed that the maximum distance between successive onset score time (*i.e.*, maximum note value) is L.

3.2.2 Musical Score Model

The musical score model represents the generative process of local keys $\mathbf{S}$ and musical notes $\mathbf{Z} = \{\mathbf{P}, \mathbf{O}\}$. More specifically, local keys $\mathbf{S}$ are sequentially generated by a Markov model and pitches $\mathbf{P}$ are then sequentially generated by another Markov model conditioned on $\mathbf{S}$ (Fig. 3.4). With an independent process, onset score times $\mathbf{O}$ are sequentially generated by a metrical Markov model [47, 48]. We henceforth omit to explicitly write the dependency on $\mathbf{Y}$ for brevity.

Model for Local Keys

To consider the relevance of adjacent local keys (*e.g.*, the local keys are likely to change infrequently), the local keys $\mathbf{S}$ are assumed to follow a first-order Markov model as follows:

$$s_0 \sim \mathrm{Categorical}(\boldsymbol{\pi}_0), \tag{3.1}$$

$$s_m \mid s_{m-1} \sim \mathrm{Categorical}(\boldsymbol{\pi}_{s_{m-1}}), \tag{3.2}$$

where $\boldsymbol{\pi}_0 \in \mathbb{R}_+^{24}$ and $\boldsymbol{\pi}_s \in \mathbb{R}_+^{24}$ are initial and transition probabilities. We write $\boldsymbol{\pi} = \boldsymbol{\pi}_{0:24}$. Given the similarities between keys (*e.g.*, relative transitions from C major would be similar to those from D major), a hierarchical Dirichlet or Pitman-Yor language model [80] with a shared prior generating key-specific priors and distributions would be useful for precise key modeling.

Model for Pitches

The pitches $\mathbf{P}$ are assumed to follow a first-order Markov model conditioned on the local keys $\mathbf{S}$ as follows:

$$p_0 \mid \mathbf{S} \sim \mathrm{Categorical}(\boldsymbol{\phi}_{s_0,0}), \tag{3.3}$$

$$p_j \mid p_{j-1}, \mathbf{S} \sim \mathrm{Categorical}(\boldsymbol{\phi}_{s_{m_j},p_{j-1}}), \tag{3.4}$$

where $\boldsymbol{\phi}_{s0} \in \mathbb{R}_+^K$ and $\boldsymbol{\phi}_{sp} \in \mathbb{R}_+^K$ are initial and transition probabilities for pitches in local key s, and m_j denotes a measure to which the onset of note j belongs. Let

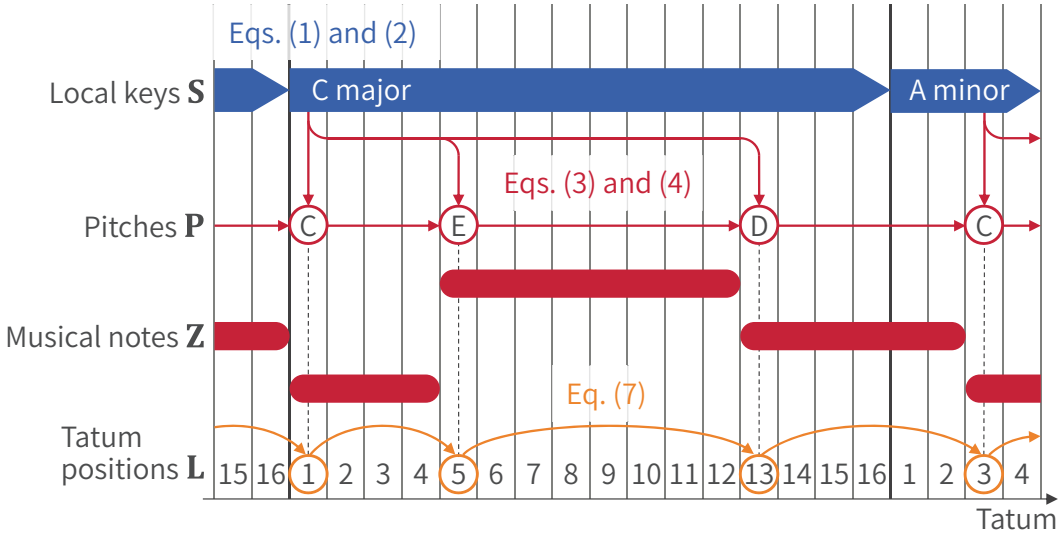


Figure 3.4: A musical score model that represents the generative process of local keys, note pitches, and note onsets. The top row represents the Markov chain of local keys. The second row represents the Markov chain of the note pitches. The third row represents a sequence of musical notes. The bottom represents the Markov chain of the onset score times of musical notes. The vertical lines represent tatum times and the bold ones represent bar lines.

$\boldsymbol{\phi} = \boldsymbol{\phi}_{1:24,0:K}$. We assume that the initial and transition probabilities in different local keys are related by a circular shift (change of tonic), and $\boldsymbol{\phi}$ are represented as follows:

$$\phi_{s0p'} \propto \bar{\phi}_{\mathrm{type}(s),0,\deg(s,p')}, \tag{3.5}$$

$$\phi_{spp'} \propto \bar{\phi}_{\mathrm{type}(s),\deg(s,p),\deg(s,p')}, \tag{3.6}$$

where $\mathrm{type}(s)\in\{\text{major, minor}\}$ indicates the type of key s, $\deg(s,p)\in\{1,\ldots,12\}$ indicates the degree of pitch p in key s (if the tonic of key s is $[s]$, $\deg(s,p)=(p-[s] \mod 12)+1$), and $\bar{\boldsymbol{\phi}}_{r0} \in \mathbb{R}_+^{12}$ and $\bar{\boldsymbol{\phi}}_{rh} \in \mathbb{R}_+^{12}$ indicate initial and transition probabilities under a local key of type $r \in \{\text{major, minor}\}$ and tonic C, where $h \in \{1,\ldots,12\}$ is a subscript representing a pitch degree. The proposed method learns only the probabilities of *relative* pitch degrees $\bar{\boldsymbol{\phi}}$ in an unsupervised or semi-supervised manner. The probabilities of *absolute* pitches $\boldsymbol{\phi}$ are then obtained by expanding $\bar{\boldsymbol{\phi}}$ according to Eqs. (3.5) and (3.6) and used for estimating a sequence of musical notes. In other words, the same transition

probabilities of pitch degrees are used for every octave range, and for pitch transitions beyond an octave we use the probabilities of the corresponding pitch transitions within an octave with the same pitch degrees.

Model for Rhythms

The onset score times $\mathbf{O}$ are assumed to follow a metrical Markov model [47,48] as follows:

$$l_{o_j} \mid l_{o_{j-1}} \sim \mathrm{Categorical}(\boldsymbol{\lambda}_{l_{o_{j-1}}}), \tag{3.7}$$

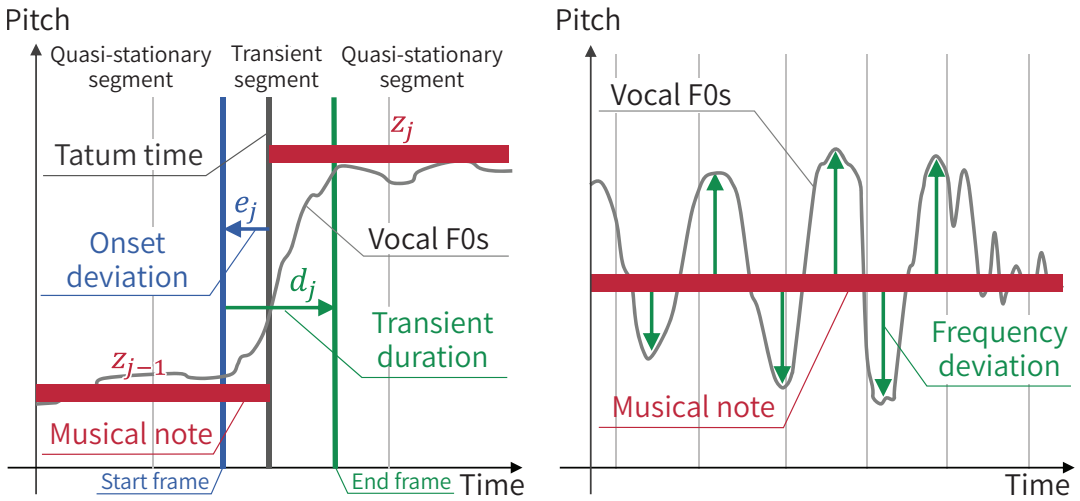
where $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{1:L} \in \mathbb{R}_+^{L \times L}$ denotes transition probabilities for tatum positions, *i.e.*, $\lambda_{l,l'}$ $(l, l' \in \{1, \ldots, L\})$ indicates the transition probability from tatum position l to l'. We interpret that if $l_{o_{j-1}} < l_{o_j}$ the onsets of notes $j-1$ and j are in the same measure and if $l_{o_{j-1}} \geq l_{o_j}$ they are in the adjacent measures.

3.2.3 F0 Trajectory Model

The F0 trajectory model represents the generative process of an F0 trajectory $\mathbf{X}$ from a note sequence $\mathbf{Z}$. We consider both temporal and frequency deviations (Fig. 3.5).

Model for Temporal Deviations

As shown in Fig. 3.5-(a), vocal F0s corresponding to each note are assumed to have a transient segment (*e.g.*, portamento) and a quasi-stationary segment (*e.g.*, vibrato). The actual onset time of note j, which is defined as the first frame of the transient segment, can deviate from the tatum time t_{o_j}. Let $e_j \in [e_{\mathrm{min}}, e_{\mathrm{max}}]$ be the deviation of the actual onset time from t_{o_j}, where $[e_{\mathrm{min}}, e_{\mathrm{max}}]$ indicates its range. The onset and offset time deviations at the start and end of the musical piece are fixed to zero ($e_1 = e_{J+1} = 0$), and the onset time deviation of the supplementary note z_0 is also set to zero for convenience ($e_0 = 0$). If $e_j < 0$ ($e_j > 0$), note j begins earlier (later) than t_{o_j}. Because $\mathbf{E} = e_{0:J}$ are considered to be distributed according to a possibly multi-modal distribution,



(a) Temporal deviations (b) Frequency deviations

Figure 3.5: Temporal and frequency deviations in vocal F0 trajectories. In both figures, the black vertical lines represent tatum times. In Fig. (a), the blue and green vertical lines represent the start and end frames of the transient segment of a vocal F0 trajectory. In Fig. (b), the arrows represent the frequency deviations of a vocal F0 trajectory from the frequency of a musical note.

in this chapter we use a categorical distribution as the most basic distribution of discrete variables as follows:

$$e_j \sim \text{Categorical}(\boldsymbol{\epsilon}), \tag{3.8}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}_+^{e_{\max}-e_{\min}+1}$ is a set of deviation probabilities.

Let $d_j \in \{1, \ldots, d_{\max}\}$ be the duration of the transient segment of note z_j, where $d_{\max}$ is the maximum number, and we set $d_0 = d_{J+1} = 1$. For the same reason as that for $\mathbf{E}$, we use a categorical distribution for $\mathbf{D} = d_{0:J}$ as follows:

$$d_j \sim \text{Categorical}(\boldsymbol{\delta}), \tag{3.9}$$

where $\boldsymbol{\delta} \in \mathbb{R}_+^{d_{\max}}$ indicates a set of duration probabilities.

Model for Frequency Deviations

As shown in Fig. 3.5-(b), the vocal F0 trajectory $\mathbf{X}$ is generated by imparting frequency deviations to a temporally deviated pitch trajectory determined by the musical notes $\mathbf{Z}$, the onset time deviations $\mathbf{E}$, and the transient durations

$\mathbf{D}$. Since vocal F0s can significantly deviate from score-indicated pitches, $\mathbf{X}$ are assumed to follow Cauchy distributions, which are more robust to outliers than Gaussian distributions, as follows:

$$x_t \mid \mathbf{Z}, \mathbf{E}, \mathbf{D} \sim \mathrm{Cauchy}(\mu_t, \sigma), \tag{3.10}$$

where μ_t and σ are the location and scale parameters, respectively. Note that if $x_t = \mathrm{uv}$, x_t is treated as missing data. The related studies [77,78] also used the Cauchy distribution for frequency deviations as a better choice than the Gaussian distribution. As shown in Fig. 3.6, the actual duration of note j is given by $[t_{o_j} + e_j, t_{o_{j+1}} + e_{j+1})$ and the reference F0 trajectory is modeled as a slanted line in the transient segment and a horizontal line in the quasi-stable segment as follows (Fig. 3.6):

$$\mu_t = \begin{cases} [p_{j-1}] + \frac{([p_j]-[p_{j-1}])(t-(t_{o_j}+e_j))}{d_j} \\ \quad (t \in [t_{o_j} + e_j, t_{o_j} + e_j + d_j)), \\ [p_j] \ \ (t \in [t_{o_j} + e_j + d_j, t_{o_{j+1}} + e_{j+1})), \end{cases} \tag{3.11}$$

where $[p_j]$ indicates a log F0 [cents] corresponding to a semitone-level pitch p_j. Although F0 transitions between different pitches have complicated dynamics in reality, in this chapter we investigate the feasibility of a simple linear transition model.

3.2.4 Bayesian Formulation

Integrating the musical score model (prior distribution of the musical notes $\mathbf{Z} = \{\mathbf{P}, \mathbf{O}\}$) with the F0 trajectory model (likelihood function of $\mathbf{Z}$ for the vocal F0s $\mathbf{X}$), we formulate an HSMM with the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\delta}, \sigma\}$ as follows:

$$\begin{aligned} p(\mathbf{X}, \mathbf{S}, \mathbf{P}, \mathbf{O}, \mathbf{E}, \mathbf{D} | \boldsymbol{\Theta}) = & \underbrace{p(\mathbf{S}|\boldsymbol{\pi}) p(\mathbf{P}|\mathbf{S}, \boldsymbol{\phi}) p(\mathbf{O}|\boldsymbol{\lambda})}_{\text{Musical score model}} \\ & \cdot \underbrace{p(\mathbf{E}|\boldsymbol{\epsilon}) p(\mathbf{D}|\boldsymbol{\delta}) p(\mathbf{X}|\mathbf{P}, \mathbf{O}, \mathbf{E}, \mathbf{D}, \sigma)}_{\text{F0 trajectory model}}, \end{aligned} \tag{3.12}$$

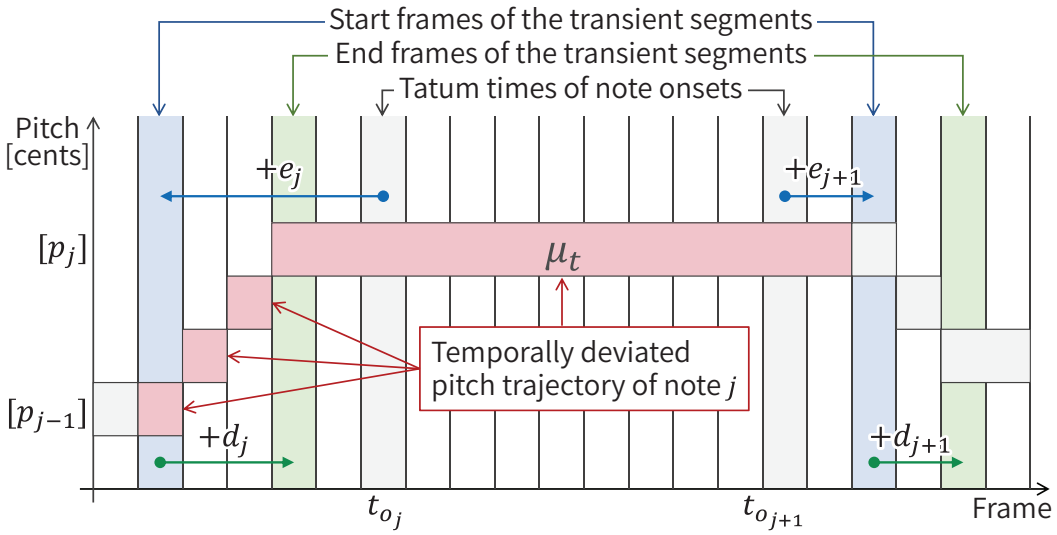


Figure 3.6: Temporally deviated pitch trajectory used as Cauchy location parameters. The blue and green vertical boxes represent the start and end frames of the transient segment and the grey vertical boxes represent the tatum times of note onsets. The red boxes represent the temporally deviated pitch trajectory of note j and the grey boxes represent the temporally deviated pitch trajectory of the other notes.

where the three terms of the musical score model are given by Eqs. (3.1) and (3.2), Eqs. (3.3) and (3.4), and Eq. (3.7), respectively, and the three terms of the F0 trajectory model are given by Eq. (3.8), Eq. (3.9), and Eq. (3.10), respectively.

We put conjugate Dirichlet priors as follows:

$$\boldsymbol{\pi}_s \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}_s^{\pi}) \quad (s \in \{0, \ldots, 24\}), \tag{3.13}$$

$$\bar{\boldsymbol{\phi}}_{rh} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}_r^{\bar{\phi}}) \quad (r \in \{\text{major, minor}\}, h \in \{0, \ldots, 12\}), \tag{3.14}$$

$$\boldsymbol{\lambda}_l \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\lambda}) \quad (l \in \{1, \ldots, L\}), \tag{3.15}$$

$$\boldsymbol{\epsilon} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\epsilon}), \tag{3.16}$$

$$\boldsymbol{\delta} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\delta}), \tag{3.17}$$

where $\boldsymbol{\gamma}_s^{\pi} \in \mathbb{R}_+^{24}$, $\boldsymbol{\gamma}_r^{\bar{\phi}} \in \mathbb{R}_+^{12}$, $\boldsymbol{\gamma}^{\lambda} \in \mathbb{R}_+^{L}$, $\boldsymbol{\gamma}^{\epsilon} \in \mathbb{R}_+^{e_{\max}-e_{\min}+1}$, and $\boldsymbol{\gamma}^{\delta} \in \mathbb{R}_+^{d_{\max}}$ are hyperparameters. We put a gamma prior on σ as follows:

$$\sigma \sim \mathrm{Gamma}(\gamma_0^{\sigma}, \gamma_1^{\sigma}), \tag{3.18}$$

where γ_0^σ and γ_1^σ are the shape and rate parameters of the gamma distribution, which are also hyperparameters.

3.3 Training and Inference

This section explains posterior inference of the latent variables and parameters. The proposed HSMM can be trained in an unsupervised manner from the vocal F0 trajectory of a target musical piece by sampling the values of the parameters and latent variables to approximate the posterior distribution of those variables (Section 3.3.1). The HSMM can also be trained in a semi-supervised manner by using the musical score model that is pretrained by using a large amount of musical scores and that is expected to learn common musical grammar and improve the musical appropriateness of the transcription results. (Section 3.3.2). The musical notes are finally estimated as the latent variables that maximize the posterior probability of them (Section 3.3.3). To obtain better results, the parameters are updated simultaneously with the latent variables.

3.3.1 Unsupervised Learning

Given an F0 trajectory $\mathbf{X}$ as observed data, our goal is to compute the posterior distribution $p(\mathbf{S}, \mathbf{P}, \mathbf{O}, \mathbf{E}, \mathbf{D}, \boldsymbol{\Theta}|\mathbf{X})$ of the latent variables (the pitches $\mathbf{P}$, onset score times $\mathbf{O}$, onset deviations $\mathbf{E}$, and transient durations $\mathbf{D}$ of musical notes with the local key $\mathbf{S}$) and the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \phi, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\delta}, \sigma\}$. Since the posterior distribution cannot be computed analytically, we use a Gibbs sampling method with efficient forward-backward procedures and a Metropolis-Hastings (MH) step. The initial values of $\mathbf{Q} = \{\mathbf{P}, \mathbf{O}, \mathbf{E}, \mathbf{D}\}$ are given by quantizing $\mathbf{X}$ on the semitone and tatum grids by the majority vote method. The initial values of $\boldsymbol{\Theta}$ are drawn from Eqs. (3.13), (3.14), (3.15), (3.16), (3.17), and (3.18). Then, the following three steps are iterated until the likelihood is fully maximized.

1. Obtain $\mathbf{S}$ from $p(\mathbf{S}|\mathbf{Q}, \boldsymbol{\Theta}, \mathbf{X})$ with forward filtering-backward sampling.
2. Obtain $\mathbf{Q}$ from $p(\mathbf{Q}|\mathbf{S}, \boldsymbol{\Theta}, \mathbf{X})$ with forward filtering-backward sampling.
3. Obtain $\boldsymbol{\Theta}$ from $p(\boldsymbol{\Theta}|\mathbf{S}, \mathbf{Q}, \mathbf{X})$ with Gibbs sampling and MH sampling.

Sampling Local Keys

In the forward step, a forward message $\alpha(s_m)$ is calculated recursively as follows:

$$\alpha(s_0) = p(p_0, s_0) = p(p_0|s_0)p(s_0) = \phi_{s_0,0,p_0}\pi_{0,s_0}, \tag{3.19}$$

$$\begin{aligned}\alpha(s_m) &= p(p_{0:j_{m+1}-1}, s_m) \\ &= \sum_{s_{m-1}} p(s_m|s_{m-1})\alpha(s_{m-1}) \prod_{j=j_m}^{j_{m+1}-1} p(p_j|p_{j-1}, s_m) \\ &= \sum_{s_{m-1}} \pi_{s_{m-1}s_m}\alpha(s_{m-1}) \prod_{j=j_m}^{j_{m+1}-1} \phi_{s_m p_{j-1} p_j},\end{aligned} \tag{3.20}$$

where j_m denotes the index of the first note in measure m.

In the backward step, the local keys $\mathbf{S}$ are sampled from a conditional distribution given by

$$p(\mathbf{S}|\mathbf{P}) = p(s_M|\mathbf{P}) \prod_{m=0}^{M-1} p(s_m|s_{m+1:M}, \mathbf{P}). \tag{3.21}$$

More specifically, local keys $s_{0:M}$ are sampled in the backward order as follows:

$$s_M \sim p(s_M|\mathbf{P}) \propto \alpha(s_M), \tag{3.22}$$

$$s_m \sim p(s_m|s_{m+1:M}, \mathbf{P}) \propto \pi_{s_m s_{m+1}}\alpha(s_m). \tag{3.23}$$

Sampling Note-Level Variables

Given the local keys $\mathbf{S}$ and the F0 trajectory $\mathbf{X}$, we aim to jointly update $\mathbf{Q} = \{\mathbf{P}, \mathbf{O}, \mathbf{E}, \mathbf{D}\}$ by using a forward filtering-backward sampling algorithm on the tatum grids. We define a forward message $\bar{\alpha}(\bar{q}_n)$ w.r.t. a tuple $\bar{q}_n = \{\bar{p}_n, \bar{o}_n, \bar{e}_n, \bar{d}_n\}$ (Fig. 3.7), where $\bar{p}_n$ and $\bar{o}_n$ indicate the pitch and onset score time of the note whose offset score time is given by n, and $\bar{e}_n$ and $\bar{d}_n$ respectively indicate the onset time deviation and transient duration of the note whose onset score time is given by n. The onset and offset times of the musical note whose offset time is n are thus given by $t_{\bar{o}_n} + \bar{e}_{\bar{o}_n}$ and $t_n + \bar{e}_n - 1$. We formally write the emission probability of F0s in this time span as follows:

$$\chi(\bar{q}_n) = \prod_{t=t_{\bar{o}_n}+\bar{e}_{\bar{o}_n}}^{t_n+\bar{e}_n-1} \mathrm{Cauchy}(x_t|\mu_t, \sigma), \tag{3.24}$$

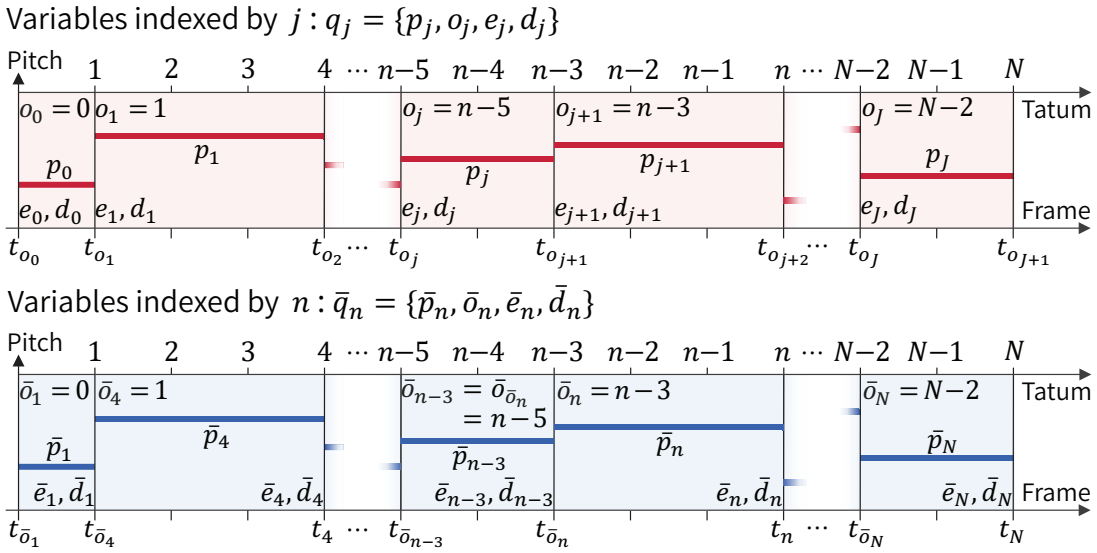


Figure 3.7: A relationship between the variables $q_j = \{p_j, o_j, e_j, d_j\}$ and $\bar{q}_n = \{\bar{p}_n, \bar{o}_n, \bar{e}_n, \bar{d}_n\}$.

where μ_t is given by the piecewise linear trajectory given by Eq. (3.11) as follows:

$$\mu_t = \begin{cases} [\bar{p}_{\bar{o}_n}] + ([\bar{p}_n] - [\bar{p}_{\bar{o}_n}])(t - (t_{\bar{o}_n} + \bar{e}_{\bar{o}_n}))/\bar{d}_{\bar{o}_n} \\ \quad (t \in [t_{\bar{o}_n} + \bar{e}_{\bar{o}_n}, t_{\bar{o}_n} + \bar{e}_{\bar{o}_n} + \bar{d}_{\bar{o}_n})), \\ [\bar{p}_n] \quad (t \in [t_{\bar{o}_n} + \bar{e}_{\bar{o}_n} + \bar{d}_{\bar{o}_n}, t_n + \bar{e}_n)). \end{cases} \tag{3.25}$$

The variable $\bar{q}_n$ is indexed by tatum n (unlike note j) to enable estimation by a hidden semi-Markov model whereby the number of notes and the onset score time of each note are obtained as a result [81].

In the forward step, a forward message $\bar{\alpha}(\bar{q}_n)$ is calculated recursively as follows:

$$\bar{\alpha}(\bar{q}_1) = p(\bar{q}_1|\mathbf{S}, \mathbf{X}) = \phi_{s_0,0,\bar{p}_1}, \tag{3.26}$$

$$\begin{aligned} \bar{\alpha}(\bar{q}_n) &= p(x_{0:t_n+\bar{e}_n-1}, \bar{q}_n|\mathbf{S}, \mathbf{X}) \\ &= \sum_{\bar{q}_{\bar{o}_n}} p(x_{0:t_n+\bar{e}_n-1}, \bar{q}_{\bar{o}_n}, \bar{q}_n|\mathbf{S}, \mathbf{X}) \\ &= p(\bar{e}_n)p(\bar{d}_n)\sum_{\bar{q}_{\bar{o}_n}} p(n|\bar{o}_n)p(\bar{p}_n|\bar{p}_{\bar{o}_n}, s_{\bar{o}_n})\chi(\bar{q}_n)\bar{\alpha}(\bar{q}_{\bar{o}_n}) \\ &= \epsilon_{\bar{e}_n}\delta_{\bar{d}_n}\sum_{\bar{q}_{\bar{o}_n}} \lambda_{l_{\bar{o}_n},l_n}\phi_{s_{\bar{o}_n},\bar{p}_{\bar{o}_n},\bar{p}_n}\chi(\bar{q}_n)\bar{\alpha}(\bar{q}_{\bar{o}_n}), \end{aligned} \tag{3.27}$$

where $s_{\bar{o}_n}$ indicates the local key of a measure including the tatum $\bar{o}_n$.

In the backward step, the variables $\mathbf{Q}$ are sampled from a conditional distribution given by

$$p(\mathbf{Q}|\mathbf{S},\mathbf{X}) = p(q_J|\mathbf{S},\mathbf{X}) \prod_{j=0}^{J-1} p(q_j|q_{j+1:J},\mathbf{S},\mathbf{X}), \tag{3.28}$$

where $q_j = \{p_j, o_j, e_j, d_j\}$ is a tuple of the semitone-level pitch, onset score time, onset time deviation, and transient duration of j-th note, respectively (Fig. 3.7). The variables $\mathbf{Q}$, however, cannot be sampled directly from Eq. (3.28) because the number of notes J is unknown before sampling notes. Instead of sampling q_j, the variable $\bar{q}_n$ is recursively sampled in the reverse order as follows:

$$\bar{q}_N \sim p(\bar{q}_N|\mathbf{S},\mathbf{X}) \propto \bar{\alpha}(\bar{q}_N), \tag{3.29}$$

$$\begin{aligned}\bar{q}_{\bar{o}_n} &\sim p(\bar{q}_{\bar{o}_n}|\bar{q}_n,\mathbf{S},\mathbf{X}) \\ &\propto \epsilon_{\bar{e}_n}\delta_{\bar{d}_n}\lambda_{\bar{o}_{\bar{o}_n},\bar{o}_n}\phi_{s_{\bar{o}_n},\bar{p}_{\bar{o}_n},\bar{p}_n}\chi(\bar{q}_n)\bar{\alpha}(\bar{q}_{\bar{o}_n}).\end{aligned} \tag{3.30}$$

As a result of the sampling, J is determined as the number of the sampled tuples.

Sampling Model Parameters

Given the latent variables $\mathbf{S}$ and $\mathbf{Q}$, the model parameters $\boldsymbol{\Theta}$ except for σ are sampled from the conditional posterior distributions as follows:

$$\boldsymbol{\pi}_s \mid \mathbf{S} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}_i^{\pi} + \mathbf{c}_i^{\pi}), \tag{3.31}$$

$$\bar{\boldsymbol{\phi}}_{rh} \mid \mathbf{S},\mathbf{P},\mathbf{O} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}_r^{\bar{\phi}} + \mathbf{c}_{rh}^{\bar{\phi}}), \tag{3.32}$$

$$\boldsymbol{\lambda}_l \mid \mathbf{O} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\lambda} + \mathbf{c}_l^{\lambda}), \tag{3.33}$$

$$\boldsymbol{\epsilon} \mid \mathbf{E} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\epsilon} + \mathbf{c}^{\epsilon}), \tag{3.34}$$

$$\boldsymbol{\delta} \mid \mathbf{D} \sim \mathrm{Dirichlet}(\boldsymbol{\gamma}^{\delta} + \mathbf{c}^{\delta}), \tag{3.35}$$

where $\mathbf{c}_s^{\pi} \in \mathbb{R}_+^{24}$, $\mathbf{c}_{rh}^{\bar{\phi}} \in \mathbb{R}_+^{12}$, $\mathbf{c}_l^{\lambda} \in \mathbb{R}_+^{L}$, $\mathbf{c}^{\epsilon} \in \mathbb{R}^{e\mathrm{max}-e_{\mathrm{min}}+1}$, and $\mathbf{c}^{\delta} \in \mathbb{R}_+^{d_{\mathrm{max}}}$ are count data obtained from $\mathbf{S}$ and $\mathbf{Q}$. More specifically, c_{0s}^{π} indicates the number of times that $s_0 = s$ is satisfied, $c_{ss'}^{\pi}$ indicates the number of transitions from key s to key s', $c_{r0h}^{\bar{\phi}}$ indicates the number of times that $\mathrm{type}(s_0) = r$ and $\deg(s_0, p_0) = h$ are both satisfied, $c_{rhh'}^{\bar{\phi}}$ indicates the number of transitions from a pitch degree h to a pitch degree h' under a key type r, $c_{ll'}^{\lambda}$ indicates the number of transitions from

a tatum position l to a tatum position l', c_e^{ϵ} indicates the number of onset time deviations taking e, and c_d^{δ} indicates the number of transient durations taking d.

To update σ, we use a MH algorithm with a random-walk proposal distribution as follows:

$$q(\sigma^*|\sigma) = \mathrm{Gamma}(\sigma^*|\sigma, 1), \tag{3.36}$$

where σ is a current sample and σ^* is a proposal. The proposal σ^* is accepted as the next sample with the probability given by

$$\mathcal{A}(\sigma^*, \sigma) = \min\left(\frac{\mathcal{L}(\sigma^*)q(\sigma|\sigma^*)}{\mathcal{L}(\sigma)q(\sigma^*|\sigma)}, 1\right), \tag{3.37}$$

where $\mathcal{L}(\sigma)$ is the likelihood function of σ given by

$$\begin{aligned}\mathcal{L}(\sigma) = &\mathrm{Gamma}(\sigma|\gamma_0^{\sigma}, \gamma_1^{\sigma}) \\ &\prod_{j=1}^{J} \prod_{t=t_{o_j}+e_{o_j}}^{t_{o_{j+1}}+e_{o_{j+1}}-1} \mathrm{Cauchy}(x_t|\mu_t, \sigma).\end{aligned} \tag{3.38}$$

3.3.2 Semi-supervised Learning

An effective way of improving the performance of AST is to estimate the parameters of the musical score model from existing musical scores (monophonic note sequences with key annotations) in advance. Let $\hat{\mathbf{S}}$ and $\hat{\mathbf{Z}} = \{\hat{\mathbf{P}}, \hat{\mathbf{O}}\}$ denote local keys, pitches, and onset score times in training data, which are defined in the same way as $\mathbf{S}$ and $\mathbf{Z} = \{\mathbf{P}, \mathbf{O}\}$ of a target piece (Section 3.2.2). Given $\hat{\mathbf{S}}$ and $\hat{\mathbf{Z}}$, the initial and transition probabilities of pitch classes $\bar{\phi}$ are obtained by normalizing the count data $\mathbf{c}_{rh}^{\bar{\phi}}$ obtained from $\hat{\mathbf{S}}$ and $\hat{\mathbf{Z}}$. Similarly, the onset transition probabilities $\boldsymbol{\lambda}$ are obtained from $\hat{\mathbf{O}}$. The initial and transition probabilities of local keys $\boldsymbol{\pi}$ are not trained because the key transitions tend to be unique to each musical piece and are learned in an unsupervised manner. Keeping $\bar{\phi}$ and $\boldsymbol{\lambda}$ fixed, the other parameters and latent variables are estimated for a target piece in a semi-supervised manner.

3.3.3 Posterior Maximization

Our final goal is to obtain the optimal values of $\mathbf{S}$, $\mathbf{Q}$, and $\mathbf{\Theta}$ that maximize the posterior probability $p(\mathbf{S}, \mathbf{Q}, \mathbf{\Theta}|\mathbf{X})$. First, we choose the best samples of $\mathbf{S}$, $\mathbf{Q}$, and $\mathbf{\Theta}$ that maximize $p(\mathbf{S}, \mathbf{Q}, \mathbf{\Theta}|\mathbf{X})$ in the Gibbs sampling described in 3.3.1. Then, the following three steps are iterated until convergence.

1. Obtain $\mathbf{S}$ that maximizes $p(\mathbf{S}|\mathbf{Q}, \mathbf{\Theta}, \mathbf{X})$ with Viterbi decoding on the upper-level chain of $\mathbf{S}$.
2. Obtain $\mathbf{Q}$ that maximizes $p(\mathbf{Q}|\mathbf{S}, \mathbf{\Theta}, \mathbf{X})$ with Viterbi decoding on the lower-level chain of $\mathbf{Q}$.
3. Obtain $\mathbf{\Theta}$ that maximizes $p(\mathbf{\Theta}|\mathbf{S}, \mathbf{Q}, \mathbf{X})$.

We empirically confirmed that a few iterations are sufficient to reach convergence. In the Viterbi algorithm in step 2 above, weighting factors β^{ϕ}, β^{λ}, β^{ϵ}, β^{δ}, and β^{χ} are introduced in the forward calculations to balance the individual sub-models. A penalization term $\exp[\beta^{o}/(o_{j+1} - o_j)]$ for long durations $o_{j+1} - o_j$ with a weighting factor β^{o} is also introduced in the forward calculations to suppress the frequent occurrence of long notes.

Estimating Local Keys

In the forward step, a Viterbi variable $\omega(s_m)$ is calculated recursively by replacing the sum operation with the max operation in the recursion of $\alpha(s_m)$ (Section 3.3.1) as follows:

$$\omega(s_0) = \phi_{s_0,0,p_0}\pi_{0,s_0}, \tag{3.39}$$

$$\omega(s_m) = \max_{s_{m-1}} \pi_{s_{m-1}s_m}\omega(s_{m-1}) \prod_{j=j_m}^{j_{m+1}-1} \phi_{s_m p_{j-1} p_j}, \tag{3.40}$$

where an argument s_{m-1} that maximizes the max operation is memorized as $\mathrm{prev}(s_m)$ when calculating $\omega(s_m)$.

In the backward step, the local keys $\mathbf{S}$ are obtained in the reverse order as follows:

$$s_M = \operatorname*{argmax}_i \omega(s_M = i), \tag{3.41}$$

$$s_m = \mathrm{prev}(s_{m+1}). \tag{3.42}$$

Estimating Musical Notes

In the forward step, a Viterbi variable $\bar{\omega}(\bar{q}_n)$ is calculated recursively by replacing the sum operation with the max operation in the recursion of $\bar{\alpha}(\bar{q}_n)$ (Section 3.3.1). In practice, we can introduce weighting factors to balance the musical score model and the F0 trajectory model, as is usually done in statistical speech recognition [82]. The modified message is thus given by

$$\bar{\omega}(\bar{q}_1) = \phi_{s_0,0,\bar{p}_1}^{\beta^\phi}, \tag{3.43}$$

$$\bar{\omega}(\bar{q}_n) = \epsilon_{\bar{e}_n}^{\beta^\epsilon} \delta_{\bar{d}_n}^{\beta^\delta} \max_{\bar{q}_{\bar{o}_n}} \lambda_{l_{\bar{o}_n},l_n}^{\beta^\lambda} \phi_{s_{\bar{o}_n},\bar{p}_{\bar{o}_n},\bar{p}_n}^{\beta^\phi} \chi(\bar{q}_n)^{\beta^\chi} \bar{\omega}(\bar{q}_{\bar{o}_n}), \tag{3.44}$$

where an argument $\bar{q}_{\bar{o}_n}$ that maximizes the max operation is memorized as $\mathrm{prev}(\bar{q}_n)$ when calculating $\bar{\omega}(\bar{q}_n)$, and β^ϕ, β^λ, β^ϵ, β^δ, and β^χ are weighting factors.

Our preliminary experiments show that the latent variables estimated with an HSMM favor longer durations for reducing the number of state transitions because the accumulated multiplication of transition probabilities reduces the likelihood. As a possible solution for penalizing longer musical notes, we introduce an additional term $f(\bar{o}_n) = \{\exp(\frac{1}{n-\bar{o}_n})\}^{\beta^o}$ to Eq. (3.44) as follows:

$$\bar{\omega}(\bar{q}_n) = \epsilon_{\bar{e}_n}^{\beta^\epsilon} \delta_{\bar{d}_n}^{\beta^\delta} \max_{\bar{q}_{\bar{o}_n}} \lambda_{l_{\bar{o}_n},l_n}^{\beta^\lambda} \phi_{s_{\bar{o}_n},\bar{p}_{\bar{o}_n},\bar{p}_n}^{\beta^\phi} \chi(\bar{q}_n)^{\beta^\chi} \bar{\omega}(\bar{q}_{\bar{o}_n}) f(\bar{o}_n), \tag{3.45}$$

where $\bar{o}_n$ and n indicate the onset and offset score times (*i.e.*, $n - \bar{o}_n$ indicates the note value) and β^o is a weighting factor.

In the backward step, the musical notes $\mathbf{Q}$ are obtained in the reverse order as follows:

$$\bar{q}_N = \operatorname*{argmax}_{\bar{q}} \omega(\bar{q}_N = \bar{q}), \tag{3.46}$$

$$\bar{q}_{\bar{o}_n} = \mathrm{prev}(\bar{q}_n). \tag{3.47}$$

Estimating Model Parameters

Given the latent variables $\mathbf{S}$ and $\mathbf{Q}$, the model parameters $\boldsymbol{\Theta}$ except for σ are obtained as the expectations of the posterior Dirichlet distributions given in

Section 3.3.1. The Cauchy scale σ is updated to a proposal given by Eq. (3.36) only when the posterior given by the product of Eqs. (3.12)–(3.18) is increased.

3.4 Evaluation

We conducted comparative experiments to evaluate the performance of the proposed method for AST. We investigated the effectiveness of the pretraining method in comparison with the unsupervised and semi-supervised learning methods (Section 3.4.2) based on the learning configurations described in Section 3.4.1. We also examined the contribution of the individual sub-models by ablating each of them (Sections 3.4.2, 3.4.2, and 3.4.2) based on the model configurations described in Section 3.4.1. To confirm the improvement of the performance of the proposed method, we conducted a comparative experiment using conventional methods (Section 3.4.3). To investigate the performance of the overall system that takes a music signal as input and outputs a musical score, we also tested the proposed method for F0 trajectories and tatum times automatically estimated from music signals (Section 3.4.3).

3.4.1 Experimental Conditions

Datasets

From the RWC Music Database [83], we used 63 popular songs in 4/4 time and that satisfy the requirements mentioned in Section 3.2.1. We verified the correctness of ground-truth annotations [84] of musical notes and beat times. For input vocal F0 trajectories, we used the ground-truth data in most experiments and estimated data obtained by the method in [8] for some experiments. In both cases, the ground-truth unvoiced regions were used to eliminate the influence of the performance of vocal activity detection (VAD). Similarly, for the 16th-note-level tatum times, we used the ground-truth data in most experiments and estimated data obtained by a neural beat tracking method [24] in some experiments. To prepare ground-truth scores used for evaluating the accuracy of transcribed notes, we used MIDI files in which the onset and offset times of

each note are adjusted to corresponding ground-truth beat times.

Hyperparameters

The Dirichlet priors on the initial key probabilities $\boldsymbol{\pi}_0$, the onset transition probabilities $\boldsymbol{\lambda}$, the onset time deviation probabilities $\boldsymbol{\epsilon}$, and the transient duration probabilities $\boldsymbol{\delta}$ given by Eqs. (3.13), (3.15), (3.16), and (3.17) were set to uniform distributions, *i.e.*, $\boldsymbol{\gamma}_0^{\pi}$, $\boldsymbol{\gamma}^{\lambda}$, $\boldsymbol{\gamma}^{\epsilon}$, and $\boldsymbol{\gamma}^{\delta}$ were set to all-one vectors. The Dirichlet priors on the key transition probabilities $\boldsymbol{\pi}_s$ ($s \in \{1, \ldots, 24\}$) given by Eq. (3.13) were set as $\boldsymbol{\gamma}_s^{\pi} = [1, 1, \cdots, 100, \cdots, 1]^{\mathrm{T}}$ (only the s-th element takes 100) to favor self-transition. The Dirichlet priors on the initial probabilities of pitch classes $\bar{\boldsymbol{\phi}}_{r0}$ given by Eq. (3.14) and the transition probabilities of pitch classes $\bar{\boldsymbol{\phi}}_{rh}$ ($r \in \{\text{major, minor}\}, h \in \{1, \ldots, 12\}$) were set as $\boldsymbol{\gamma}_{\text{major}}^{\bar{\phi}} = [10, 1, 10, 1, 10, 10, 1, 10, 1, 10, 1, 10]^{\mathrm{T}}$ and $\boldsymbol{\gamma}_{\text{minor}}^{\bar{\phi}} = [10, 1, 10, 10, 1, 10, 1, 10, 10, 1, 10, 1]^{\mathrm{T}}$ to favor the seven pitch classes on the C major and minor scales, respectively. The gamma prior on σ in Eq. (3.18) was set as $a_0^{\sigma} = a_1^{\sigma} = 1$. Assuming that keys tend to change infrequently, the s-th element of $\boldsymbol{\gamma}_s^{\pi}$ was set to a large value (100). Because non-diatonic notes are often used, $\boldsymbol{\gamma}_{\text{major}}^{\bar{\phi}}$ and $\boldsymbol{\gamma}_{\text{minor}}^{\bar{\phi}}$ were set to smaller values (10). Optimization of these hyperparameters is left as future work.

For a model M3 in Table 3.1 the weighting factors β^{ϕ}, β^{λ}, and β^{χ} were determined by Bayesian optimization [79] as $\beta^{\phi} = 18.9$, $\beta^{\lambda} = 49.6$, and $\beta^{\chi} = 5.1$. The weighting factors β^{ϵ} and β^{δ} were determined by grid search and set as $\beta^{\epsilon} = 20.0$ and $\beta^{\delta} = 10.0$. The weighting factor β^{o} of the duration penalty term was set to $\beta^{o} = 50$, which was experimentally selected from $\{1, 5, 10, 50, 100, 500, 1000\}$ so that the performances of M3 and M4 were maximized. Since the forward-backward algorithms (Appendices 3.3.1 and 3.3.3) are defined in a huge product space $\bar{q}_n = \{\bar{p}_n, \bar{o}_n, \bar{e}_n, \bar{d}_n\}$, the range of pitches considered was limited as follows:

$$\bar{p}_n \in \bigcup_{i=n-1}^{n+1} \left\{ p_i^{\text{Maj}} - 1, p_i^{\text{Maj}}, p_i^{\text{Maj}} + 1 \right\}, \tag{3.48}$$

where p_n^{Maj} is the pitch estimated by the majority vote method between tatums $n-1$ and n. The pitch-range constraint might prevent the proposed method from estimating some correct notes. However, it is difficult to recover the correct notes

from an erroneous F0 trajectory that is far from the ground-truth pitch sequence. The pitch-range constraint is thus effective for reducing the computational complexity of the proposed method without much damaging the performance of note estimation.

Learning Configurations

The unsupervised and semi-supervised schemes (Section 3.3.1 and Section 3.3.2) were used for estimating the initial and transition probabilities of pitch classes $\bar{\phi}$ and the onset transition probabilities $\boldsymbol{\lambda}$. In the unsupervised scheme $\bar{\phi}$ and/or $\boldsymbol{\lambda}$ were learned from only the vocal F0 trajectory of a target song. In the semi-supervised scheme $\bar{\phi}$ and/or $\boldsymbol{\lambda}$ were estimated as follows:

[L1] Pitch transition learning: $\bar{\phi}$ were learned from 90 popular songs with no overlapped sung notes except for a target song in the RWC Music Database [83].

[L2] Onset transition learning: $\boldsymbol{\lambda}$ were estimated in advance from a corpus of rock music [85].

Model Configurations

The four main components of the proposed method, *i.e.*, the local key model (Section 3.2.2), rhythm model (Section 3.2.2), temporal deviation model (Section 3.2.3), and note duration penalty (Section 3.3.3) were evaluated separately. More specifically, we examined the performance of AST when each component was included or not included as follows:

[T1] Key modeling: When this function was enabled, the key transition probabilities $\boldsymbol{\pi}$ were estimated. Otherwise, the number of local keys was set to 1, *i.e.*, $\pi_{0,1} = \pi_{1,1} = 1$.

[T2] Rhythm modeling: When this function was enabled, the onset transition probabilities $\boldsymbol{\lambda}$ were estimated. Otherwise, $\boldsymbol{\lambda}$ were set to the same value, *i.e.*, $\lambda_{l,l'} = 1$.

[T3] Temporal deviation modeling: When this function was enabled, the onset

deviations $\mathbf{E}$ and the transient durations $\mathbf{D}$ were estimated and we set $[e_{\text{min}}, e_{\text{max}}] = [-5, 5]$ and $d_{\text{max}} = 10$. Otherwise, $\mathbf{E}$ and $\mathbf{D}$ were not considered, *i.e.*, $e_j = 0$ and $d_j = 1$.

[T4] Note duration penalization: When this function was enabled, the penalty term $f(o_n)$ was introduced in the Viterbi decoding. Otherwise, $f(o_n)$ was not used.

Evaluation Measures

The performance of AST was evaluated in terms of the frame-level and note-level measures. The frame-level accuracy, $\mathcal{A}$, is defined as follows:

$$\mathcal{A} = \frac{T_{\text{correct}}}{T_{\text{gt}}}, \tag{3.49}$$

where T_{gt} is the total number of voiced frames in the beat-adjusted MIDI files and T_{correct} is the total number of voiced frames whose semitone-level pitches were correctly estimated. The note-level measures including the precision rate $\mathcal{P}$, the recall rate $\mathcal{R}$, and the F-measure $\mathcal{F}$ are defined as follows:

$$\mathcal{P} = \frac{J_{\text{correct}}}{J_{\text{output}}}, \quad \mathcal{R} = \frac{J_{\text{correct}}}{J_{\text{gt}}}, \quad \mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{3.50}$$

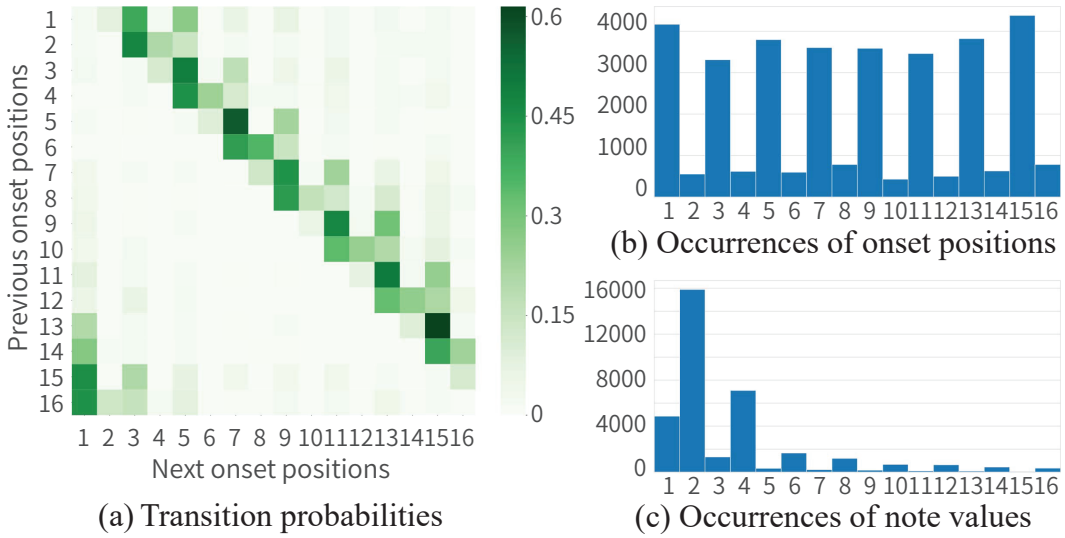
where J_{gt} is the total number of musical notes in the ground-truth scores, J_{output} is the total number of musical notes in the transcribed scores, and J_{correct} is the total number of correctly-estimated musical notes. We used three criteria for judging that a musical note is correctly estimated [86]:

[C1] Pitch, onset, and offset match: The semitone-level pitch and the onset and offset score times were all estimated correctly.

[C2] Pitch and onset match: The semitone-level pitch and the onset score time were estimated correctly.

[C3] Onset match: The onset score time was estimated correctly.

When the tatum times were estimated with the beat tracking method [24], we used error tolerance for onset and offset times to absorb slight differences between the estimated and ground-truth tatum times. More specifically, an esti-



(a) Transition probabilities

(b) Occurrences of onset positions

(c) Occurrences of note values

Figure 3.8: Pretrained transition probabilities between the 16th-note-level tatum positions.

mated onset time was judged as correct if it was within 50 ms from the ground-truth onset time. An estimated offset time was judged as correct if it was within 50 ms or 20% of the duration of the ground-truth note from the ground-truth offset time.

3.4.2 Experimental Results

The experimental results obtained with the different learning and model configurations are shown in Table 3.1. In this experiment, the ground-truth vocal F0 trajectories and tatum times were given as input to the proposed method.

Learning Configurations

We evaluated the effectiveness of the semi-supervised learning strategy by comparing the performances for the four different learning configurations M1, M2, M3, and M4 in Table 3.1. The accuracies for M3 (73.5%) and M4 (73.7%) were better than those for M1 (71.4%) and M2 (71.5%). This indicates that the pretraining of the onset transition probabilities $\boldsymbol{\lambda}$ was effective for improving the performance of AST. Examples of $\boldsymbol{\lambda}$ are shown in Fig. 3.8. The probabilities of transitions to the 8th-note-level tatum grids were larger than those to the other

Table 3.1: Performance of the proposed method with different learning and model configurations.

	L1 Pitch trans. learning	L2 Onset trans. learning	T1 Key modeling	T2 Rhythm modeling	T3 Temp. dev. modeling	T4 Note dur. penalization
M1			✓	✓		
M2	✓		✓	✓		
M3		✓	✓	✓		
M4	✓	✓	✓	✓		
M5						
M6			✓			
M7		✓		✓		
M8		✓	✓	✓	✓	
M9		✓	✓	✓		✓
M10		✓	✓	✓	✓	✓
M11	✓	✓	✓	✓	✓	✓

		C1 Pitch, onset, offset			C2 Pitch, onset			C3 Onset		
	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
M1	71.4	21.4	16.6	18.6	40.7	30.5	34.6	52.8	39.1	44.6
M2	71.5	21.3	16.3	18.3	41.0	30.3	34.6	53.7	39.3	45.0
M3	73.5	**28.7**	24.0	26.0	**47.1**	38.9	42.3	**62.6**	51.5	56.2
M4	73.6	28.2	23.3	25.3	**47.1**	38.4	42.0	**62.6**	51.0	55.8
M5	71.7	21.7	19.2	20.2	39.9	35.1	37.1	55.2	48.8	51.5
M6	72.9	22.2	19.8	20.8	40.8	36.1	38.0	55.0	49.0	51.5
M7	72.6	27.7	22.9	24.9	46.2	37.6	41.2	62.5	50.8	55.6
M8	73.5	27.3	23.8	25.3	45.5	39.2	41.9	61.2	52.8	56.3
M9	**73.7**	28.4	**26.5**	**27.3**	44.6	41.3	**42.6**	61.3	57.0	58.7
M10	73.1	27.5	26.4	26.8	43.8	**41.9**	**42.6**	60.5	**58.1**	**58.9**
M11	73.1	27.1	25.1	25.9	44.1	40.6	42.0	60.8	56.3	58.1

positions (Figs. 3.8-(a) and 3.8-(b)) and the 8th notes frequently appeared in the date used for the pretraining (Fig. 3.8-(c)). This enables the proposed method to output scores with natural rhythms.

The accuracy for M1 (71.4%) was almost the same as that for M2 (71.5%) and that for M3 (73.5%) was almost the same as that for M4 (73.6%). This

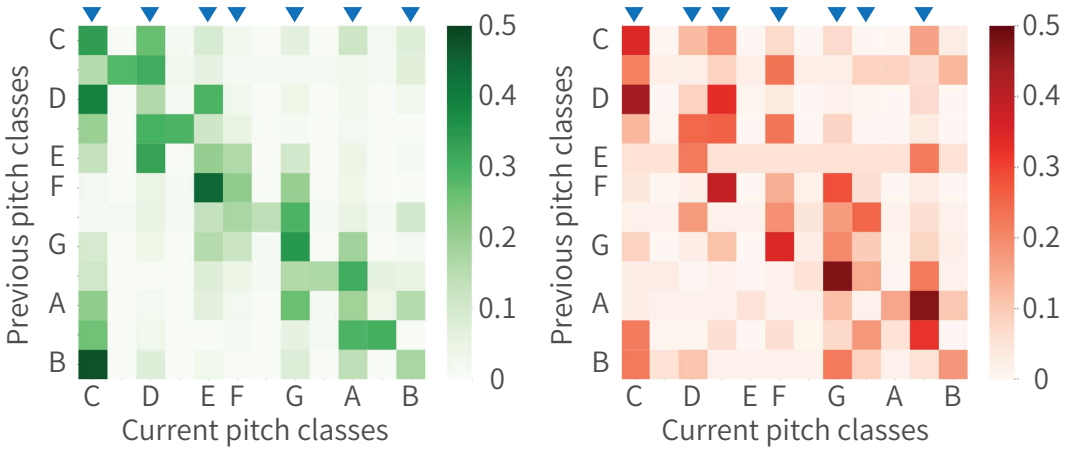
Table 3.2: Performance of the conventional and proposed methods.

		C1			C2			C3		
	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
Majority-vote	54.0	10.1	19.1	13.1	21.2	42.6	28.0	37.3	77.7	49.9
BS-HMM [87]	64.2	9.3	9.4	9.2	23.9	23.7	23.5	37.6	37.7	37.2
M3 (proposed)	**73.5**	**34.7**	**32.4**	**33.3**	**49.6**	**46.1**	**47.4**	**64.5**	59.9	61.6

Table 3.3: Performance of the proposed method based on E0 estimation and/or tatum detection.

				C1			C2			C3		
	F0	Tatums	$\mathcal{A}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
M3	[84]	[84]	**73.5**	**28.7**	24.0	**26.0**	**47.1**	38.9	**42.3**	**62.6**	51.5	**56.2**
M3-1	[8]	[84]	73.0	28.1	**24.4**	25.9	46.4	**39.7**	42.5	61.3	**52.5**	56.1
M3-2	[84]	[24]	71.2	25.7	21.5	23.3	43.5	35.9	39.0	58.3	47.8	52.1
M3-3	[8]	[24]	70.7	25.9	22.7	24.1	43.6	37.7	40.1	57.8	49.8	53.2

indicates that the pretraining of the transition probabilities of pitch classes $\bar{\phi}$ did not much improve the performance of AST. The precision rates, recall rates, and F-measures for M2 and M4 were slightly worse than those for M1 and M3 especially in terms of C1, respectively. As shown in Fig. 3.9, when the transition probabilities of pitch classes were pretrained, pitches were likely to continue or change to near pitches. As shown in Figs. 3.10 and 3.11, in contrast, there was no clear bias in the transition probabilities estimated by the unsupervised learning method or the posterior maximization method because of the effect of the prior distribution. Estimation errors caused by using the pretrained transition probabilities are shown in Fig. 3.12, where the ground-truth key was D minor in both examples. In the example 1, M3 failed to estimate the correct key, but succeeded in estimating the correct notes on the scale of the estimated key. In contrast, M4 failed to estimate the correct key and two estimated notes were out of the scale of the estimated key. M4 also failed to estimate even the note on the scale of the estimated key because vocal F0s corresponding to the note were represented as frequency deviations. In the example 2, M3 succeeded in estimating the correct note on the scale of the incorrectly estimated key.



(a) Transition probabilities between pitch classes in C major

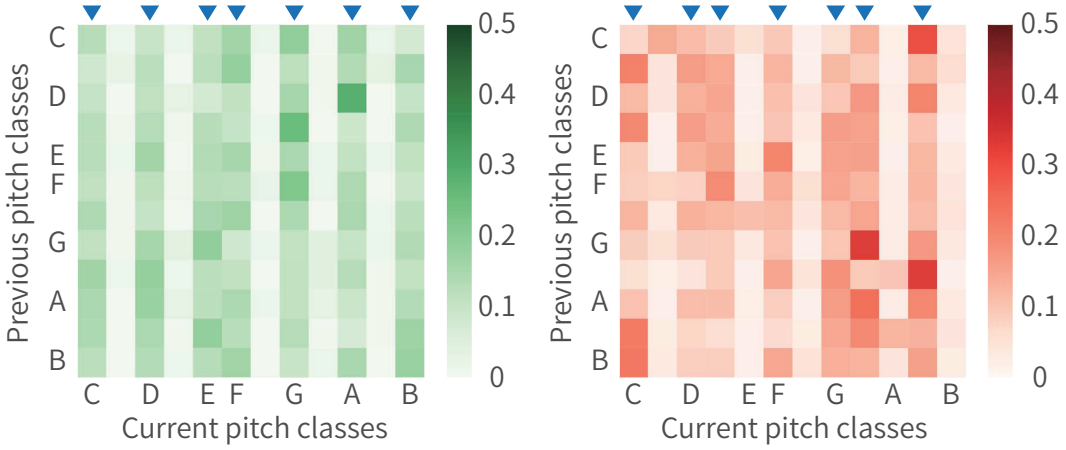
(b) Transition probabilities between pitch classes in C minor

Figure 3.9: Pretrained transition probabilities between the 12 pitch classes under the major and minor diatonic scales.

M4 estimated a key as F major (a relative major key of D minor), but incorrectly estimated the note because vocal F0s corresponding to the note were represented as frequency deviations. As in these examples, incorrect key estimation and/or incorrectly represented F0s sometimes led to incorrect note estimation.

Key and Rhythm Modeling

We evaluated the effectiveness of the key and rhythm modeling by comparing the performances for M3, M5, M6, and M7 in Table 3.1. Note that the pretrained onset transition probabilities $\boldsymbol{\lambda}$ were not used for M5 and M6 that do not model rhythms. The accuracy for M3 (73.6%) was better than those for M5 (71.7%), M6 (72.9%), and M7 (72.6%). This indicates that the musical score model including the key and rhythm models was effective for improving the performance of AST. While the accuracy for M6 (72.9%) was almost the same as that for M7 (72.6%), the precision rates, recall rates, and F-measures for M7 were much better than those for M6. This indicates that the rhythm model was more effective than the key model for AST.



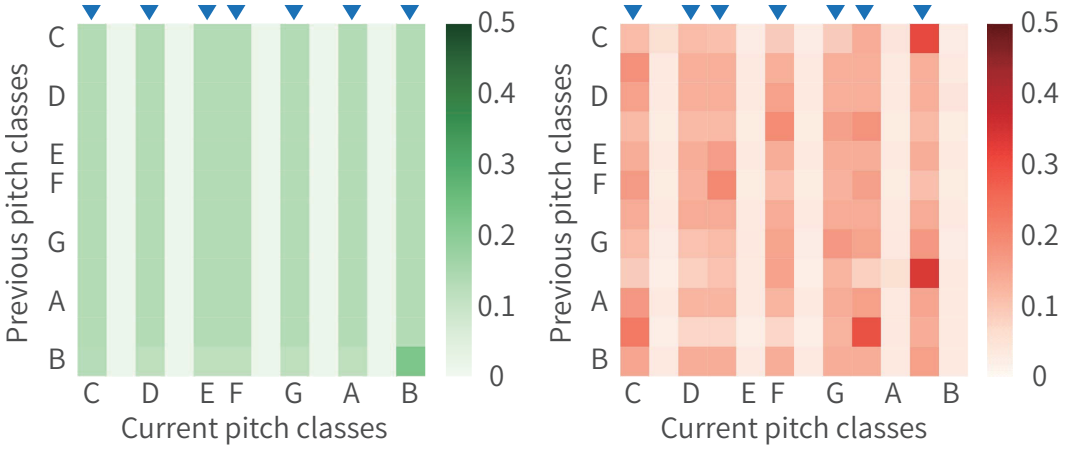
(a) Transition probabilities between pitch classes in C major

(b) Transition probabilities between pitch classes in C minor

Figure 3.10: Transition probabilities between the 12 pitch classes estimated by the unsupervised learning method (Section 3.3.1).

Temporal Deviation Modeling

We evaluated the effectiveness of the temporal deviation modeling by comparing the performances for M3 and M8 in Table 3.1. In most criteria, the performances for M8 were worse than those for M3. This was mainly due to the large deviations of a vocal F0 trajectory that were hard to represent precisely. The positive effect of the temporal deviation modeling was shown in Fig. 3.13. In these pieces, M8 successfully represented the transient segments of vocal F0s. In the left piece, M8 correctly estimated a musical note regarded as frequency deviations by M3. The temporal deviation model of M8, however, often degraded the performance, as shown in Fig. 3.14. In the top pieces, the temporal deviation model overfit *overshoot* and *preparation* in vocal F0 trajectories. Although we had expected the overshoot and preparation to be captured as frequency deviations following the Cauchy distribution, the overfitting, in fact, caused inaccurate pitch estimation. In the bottom-left piece, an extra note was inserted because the transient segment longer than the tatum interval was represented by the temporal deviations of the two successive tatums. In the lower-right piece, the proposed method failed to detect the correct onset because the transition segment was represented as the temporal deviation of the previous tatum before the correct tatum. Treatment



(a) Transition probabilities between pitch classes in C major

(b) Transition probabilities between pitch classes in C minor

Figure 3.11: Transition probabilities between the 12 pitch classes estimated by the posterior maximization method (Section 3.3.3).

of such slower F0 transitions should be included in future work.

As shown in Fig. 3.15, the categorical distribution of the onset time deviations $\mathbf{E}$ and that of the transient durations $\mathbf{D}$ estimated by the unsupervised learning method (Section 3.3.1) were found to have complicated shapes. In future work, flexible parametric models (*e.g.*, Poisson mixture models) that have fewer parameters than the categorical distribution could be used for improving the efficiency of training the distributions of the temporal deviations $\mathbf{E}$ and $\mathbf{D}$.

Note Duration Penalization

We evaluated the effectiveness of introducing the penalty term $f(o_n)$ by comparing the performances for the two different models M3 and M9 in Table 3.1. The accuracy for M9 (73.7%) was slightly better than that for M9 (73.5%). In terms of C1, the precision rate for M9 (28.4%) was slightly worse than that for M3 (28.7%), whereas the recall rate for M9 (26.5%) was better than that for M3 (24.0%). As shown in Fig. 3.16, M9 successfully estimated shorter notes that were not detected by M3. On the other hand, M9 mistakenly split longer notes correctly estimated by M3 into several shorter notes. A possible way of mitigating this trade-off problem is to extend the proposed model to deal with both

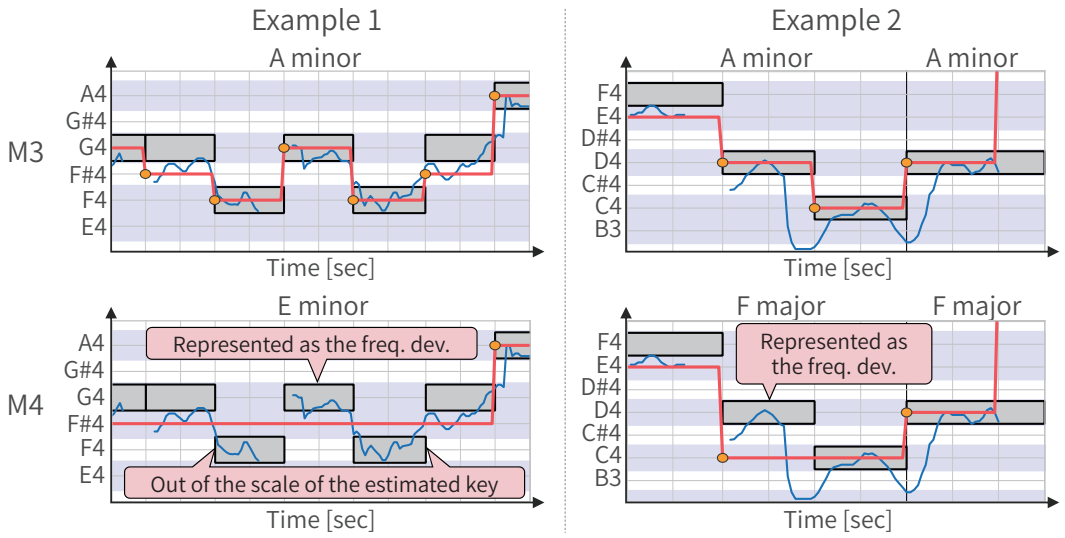


Figure 3.12: Estimation errors caused by using the pretrained initial and transition probabilities of pitch classes. The pale blue backgrounds indicate the diatonic scales of estimated keys and the gray boxes indicate ground-truth musical notes. The blue and red lines indicate vocal F0s and estimated musical notes, respectively. The orange dots indicate estimated note onsets. The gray grids indicate tatum times and semitone-level pitches. The red balloons indicate the ground-truth notes that the proposed method failed to estimate. The estimated keys are illustrated in the figure, and the ground-truth key in both examples is D minor.

pitch and volume trajectories of singing voice. This is expected to improve the performance of onset detection.

We evaluated the effectiveness of jointly using the temporal deviation modeling and the note duration penalization by comparing the performances for M9 an M10 in Table 3.1. As described in Section 3.4.2, the temporal deviation modeling decreased the accuracy from 73.7% (M9) to 73.1% (M10). We also checked the effectiveness of introducing both models into M4. Although the accuracy for M11 and that for M10 were same, the precision and recall rates and F-measures for M11 were slightly worse than those for M10. This indicates that the pretrained probabilities $\bar{\phi}$ have little effect when using the temporal deviation modeling and note duration penalization.

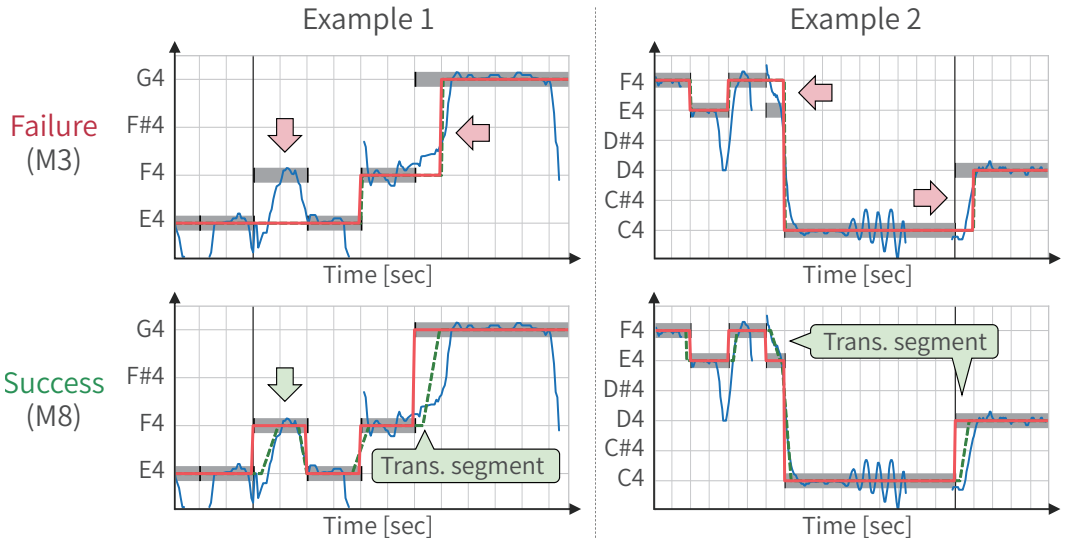


Figure 3.13: Positive effects of temporal deviation modeling (cf. Fig. 3.12). The green lines estimated F0s with temporal deviations. The red arrows indicate estimation errors and the green arrows and balloons indicate correct notes obtained by modeling temporal deviations.

3.4.3 Further Investigations

Comparison with Conventional Methods

We compared the performance obtained by M3 with those obtained by the straightforward majority-vote method and a statistical method based on a beat-synchronous HMM (BS-HMM) [87]. Since these conventional methods can only estimate a semitone-level pitch at each tatum interval, successive tatum intervals were concatenated to form a single note if they had the same pitch. Similarly, successive notes included in the ground-truth data and those estimated by the proposed method were concatenated to form a single note if they had the same pitch. To deal with unvoiced regions in vocal F0 trajectories, the BS-HMM was extended in the same way as the proposed method. Note that the majority-vote method can handle the unvoiced regions and estimate rests for those regions.

Experimental results are shown in Table 3.2. The accuracy obtained by the proposed method (73.5%) was better than those obtained by the majority-vote method (54.0%) and the BS-HMM (64.2%). Considering that the use of the musical score model is the main difference between the proposed method and

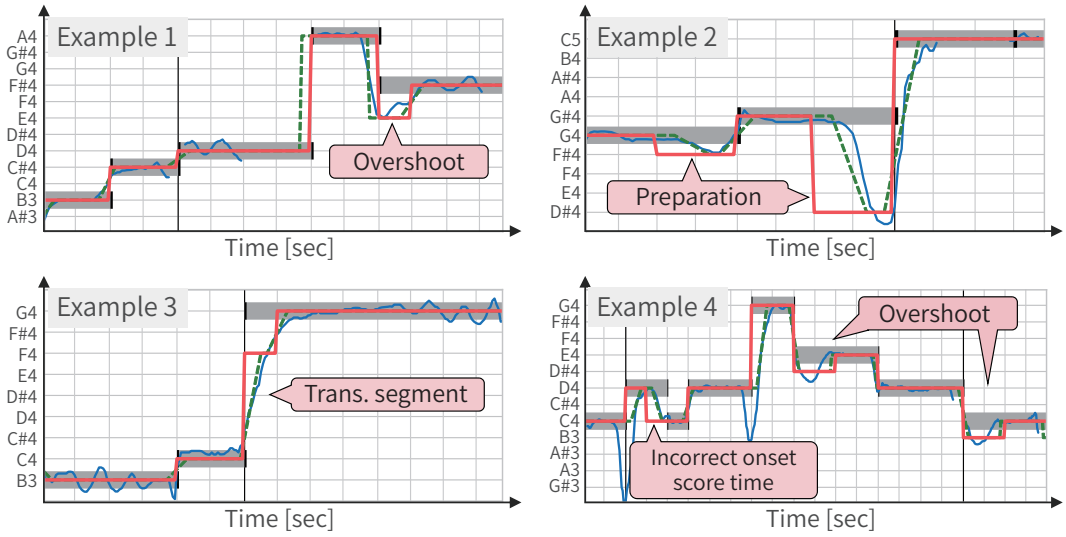


Figure 3.14: Negative effects of temporal deviation modeling (cf. Fig. 3.13). The red balloons indicate estimation errors.

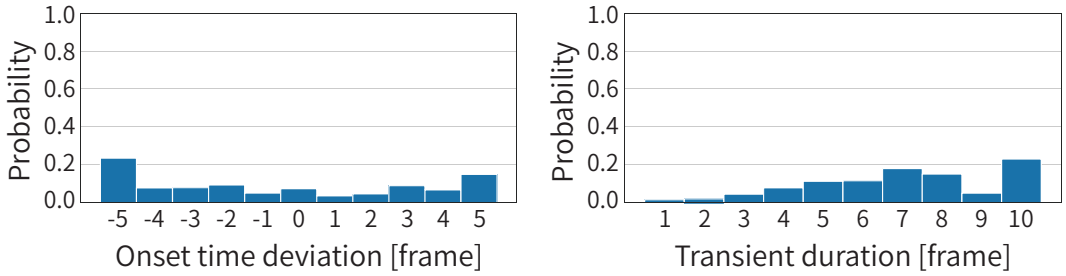


Figure 3.15: The categorical distribution of the onset time deviations $\mathbf{E}$ and that of the transient durations $\mathbf{D}$ estimated in the unsupervised learning method (Section 3.3.1).

the others, we can say that the musical score model is effective for improving the performance of AST. In terms of C3, only the recall rate obtained by the majority-vote method (77.7%) was better than those obtained by the other methods because many extra short notes were obtained.

Impacts of F0 Estimation and Tatum Detection

To evaluate the impacts of vocal F0 estimation and tatum time detection on the performance of AST, we tested M3 by using the ground-truth and estimated data as follows:

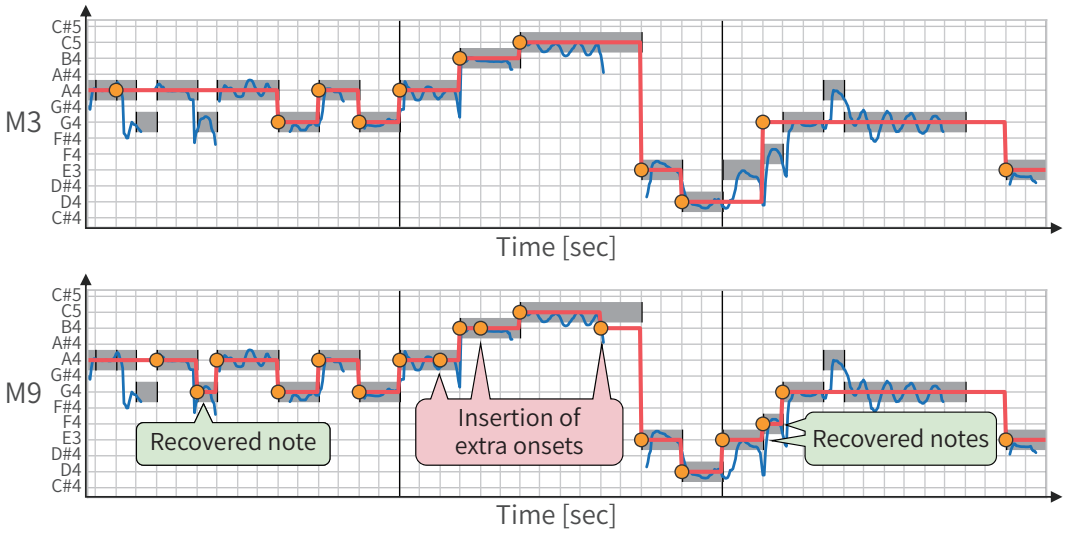


Figure 3.16: Positive and negative effects of duration penalization (cf. Fig. 3.12). The green and red balloons indicate improved and deteriorated parts.

- *F0 trajectory*: We used the ground truth data in [84] and estimated data obtained by [8].
- *Tatum times*: We used the ground truth data in [84] and estimated data detected by [24].

As shown in Table 3.3, the accuracy obtained by M3 (73.5%) using the ground-truth data was better than those obtained by M3-1 (73.0%), M3-2 (71.2%), and M3-3 (70.7%) using the estimated data. The impact of tatum detection was larger than that of F0 estimation and the F0 estimation slightly degraded the performance of the proposed method. To avoid such negative effects, we plan to develop a method that directly and jointly estimates a sequence of musical notes with tatum times from music audio signals.

3.5 Summary

This chapter presented a statistical AST method that estimates a sequence of musical notes from a vocal F0 trajectory. Our method is based on an HSMM that combines a musical score model with a vocal F0 model to represent the hierarchical generative process of a vocal F0 trajectory from a note sequence determined

by musical knowledge. We derived an efficient Bayesian inference algorithm based on the Gibbs and MH samplings for solving the inverse problem. The experimental results showed that the key and rhythm models included in the musical score model were effective for improving the performance of AST. The temporal deviation model was not always effective because it was often difficult to discriminate temporal deviations and frequency deviations.

Chapter 4

Hybrid Approach Based on CRNN-HSMM

This chapter presents the hybrid approach to AST for audio recordings of popular music consisting of monophonic singing voice and accompaniment sounds (Fig. 4.1). To directly handle the music audio signals and improve the AST performance, we use *discriminative* acoustic model based on the CRNN instead of the *generative* acoustic model in Chapter 3.

4.1 Introduction

Using a musical language model that incorporates prior knowledge on symbolic musical scores has been shown to be effective for avoiding out-of-scale pitches and irregular rhythms caused by the considerable pitch and temporal variation of the singing voice. this problem. Graphical models [31, 88–90] have been studied to integrate a language model with an acoustic model describing the generative process of acoustic features or F0s. In particular, the current state-of-the-art method [90] for audio-to-score AST is based on a hidden semi-Markov model (HSMM) consisting of a semi-Markov language model describing the generative process of a note sequence and a Cauchy acoustic model describing the generative process of an F0 contour from the musical notes. While being more accurate than other methods, the output scores include errors caused by the preceding F0 estimation step, and repeated notes of the same pitch cannot be

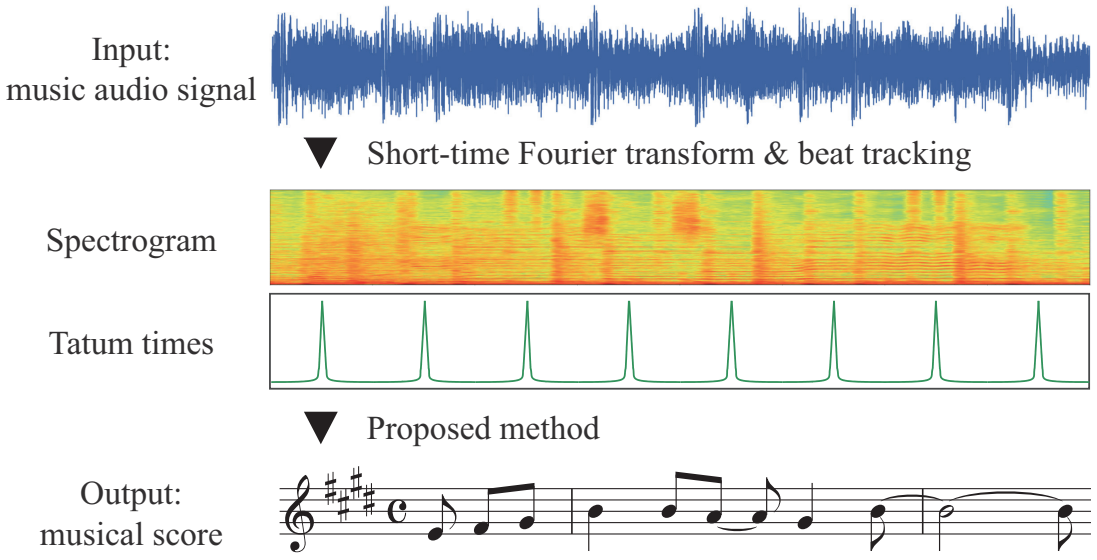


Figure 4.1: The problem of automatic singing transcription. The proposed method takes as input a spectrogram of a target music signal and tatum times and estimates a musical score of a sung melody.

detected from only F0 information. An alternative approach to AST is to use an end-to-end DNN framework to directly convert a sequence of acoustic features into a sequence of musical symbols. At present, however, this approach covers only constrained conditions (*e.g.*, the use of synthetic sound signals) and has only limited success [53, 56, 91, 92].

To solve this problem, we propose an AST method that integrates a language model with a DNN-based acoustic model. This approach can utilize both the statistical knowledge about music notes and the capacity of DNNs for describing complex data distributions of input music signals. This is known as the hybrid approach, which has been one of the major approaches to automatic speech recognition in recent years [93]. To our knowledge, the hybrid approach has not been attempted for audio-to-score AST in the literature. The language model is constructed based on a semi-Markov model (SMM) describing the generative process of local keys, note pitches, and onset times. The acoustic model describes the generative process of a music audio signal from musical notes and is constructed based on a convolutional recurrent neural network (CRNN) estimating the pitch and onset probabilities for each tatum. Since the

accuracy of beat tracking is already high [24], we assume that tatum times are estimated in advance and use them to extract tatum-level features from a frame-level audio spectrogram in the CRNN. To simplify the problem, we focus on songs in 4/4 time, which are the most typical cases in popular music. We also investigate how the application of singing voice separation for an input signal affects the transcription results.

The main contributions of this study are as follows. We propose the first DNN-HMM-type hybrid model for audio-to-score AST. Despite the active research on AST-related tasks like singing voice separation and F0 estimation, a full AST system that can output musical scores in a human-readable form has scarcely been studied. Our system can deal with polyphonic music signals and output symbolic musical scores in the MusicXML format. We found that the proposed method outperformed the HSMM-based method [90] by a large margin. We also confirmed that the language model significantly improves the AST performance, especially in the rhythm aspects. Finally, we found that the application of singing voice separation to the input music signals can further improve the performance.

The rest of this chapter is as follows: Section 4.2 describes our approach to AST. Section 4.3 reports the experimental results. Section 4.4 concludes the chapter.

4.2 Method

We specify the audio-to-score AST problem in Section 4.2.1 and describe the proposed generative modeling approach to this problem in Section 4.2.2. We formulate the CRNN-HSMM model in Sections 4.2.3 and 4.2.5. We explain how to train the model parameters in Section 4.2.6 and the transcription algorithm in Section 4.2.7.

4.2.1 Problem Specification

We formulate the audio-to-score AST problem under two simplified but practical conditions; 1) The time signature of a target song is $4/4$; and 2) the tatum times of the song, which form the 16th note-level grids, are estimated in advance (*e.g.*, [24]).

The inputs to the method are an audio spectrogram and tatum times of a given song. Similarly to the harmonic constant-Q transform (HCQT) representation [9], the audio spectrogram is obtained by stacking H pitch-shifted (upshifted and downshifted) versions of a log-frequency magnitude spectrogram obtained by warping the linear frequency axis of the short-time Fourier transform (STFT) spectrogram into the log-frequency axis. Thus, the input audio spectrogram can be represented as a tensor $\mathbf{X} \in \mathbb{R}^{H \times F \times T}$, where H, F, and T represent the number of channels, that of frequency bins, and that of time frames, respectively.

We use the notation "$_{i:j}$" to represent a sequence of integer indices from i to j. The tatum times can be represented by a sequence of frame indices $t_{1:N+1}$, where ns label tatums, N is the number of tatums in the input song, and t_{N+1} indicates a "sentinel" frame of the song. By trimming off unimportant frames before the first tatum and after the last tatum if necessary, we can assume that $t_1 = 1$ and $t_{N+1} = T + 1$. The relative position of tatum n in a measure is called the *metrical position* and denoted by $l_n \in \{1, \ldots, L\}$ ($L = 16$ is the number of tatums in each measure); $l_n = 1$ means that tatum n is a downbeat (the first tatum of a measure). In general, we have $l_{n+1} - l_n \equiv 1 \pmod{L}$; however, we do not assume $l_1 = 1$. We use the symbol $m \in \{1, \ldots, M\}$ to label measures (M is the number of measures).

The output of the proposed method is a sequence of musical notes represented by pitches $p_{1:J}$ and onset (score) time in tatum units $n_{1:J+1}$. We use the symbol j to label musical notes, and J represents the number of estimated musical notes. The pitch p_j of the jth note takes a value in $\{0, 1, \ldots, K\}$; $p_j = 0$ means that it is a rest and $p_j > 0$ means that it is a pitched note (K be the number of unique semitone-level pitches considered). The onset time n_j of the jth note takes a value in $\{1, \ldots, N + 1\}$ and, for convenience, we assume that $n_1 = 1$ and

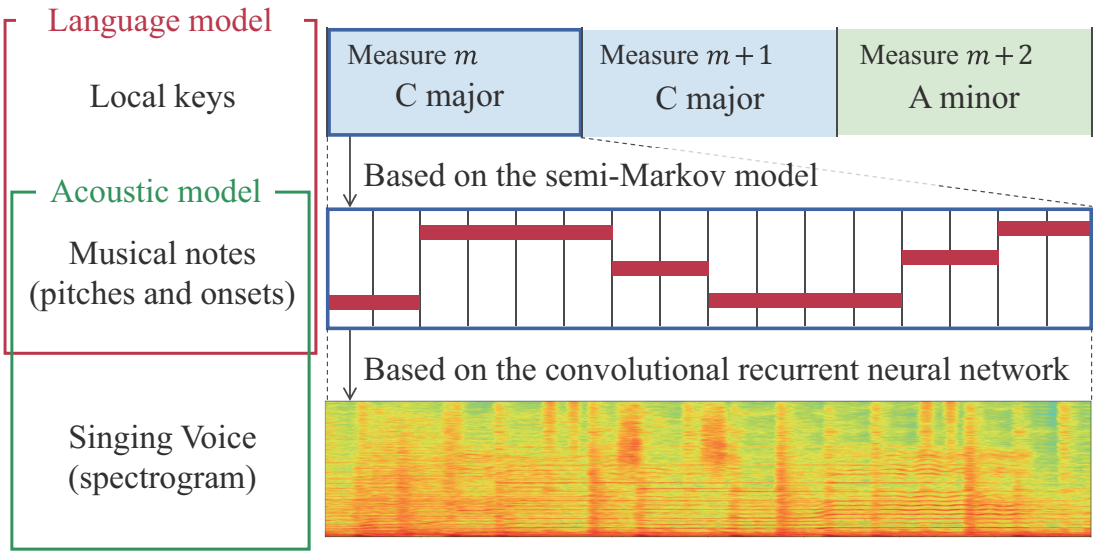


Figure 4.2: The proposed hierarchical probabilistic model that consists of a SMM-based language model representing the generative process of musical notes from local keys and a CRNN-based acoustic model representing the generative process of an observed spectrogram from the musical notes. We aim to infer the latent notes and keys from the observed spectrogram.

$n_{J+1} = N+1$. The $(J+1)$ th note onset is introduced only for defining the length of the J th note and is not used in the output transcribed score.

For musical language modeling, we introduce musical key variables, which are also estimated in the transcription process. To allow modulations (key changes) within a song, we introduce a local key s_m for each measure m. Each variable s_m takes a value in $\{1 = \mathrm{C}, 2 = \mathrm{C}\sharp, \ldots, 12 = \mathrm{B}\}$. Since similar musical scales are used in relative major and minor keys, they are not distinguished here. For example, $s_m = 0$ means that measure m is in the C major key or the A minor key.

4.2.2 Generative Modeling Approach

We propose a generative modeling approach to the audio-to-score AST problem (Fig. 4.2). We formulate a hierarchical generative model of the local keys $\mathbf{S} = s_{1:M}$, the pitches $\mathbf{P} = p_{1:J}$ and onset times $\mathbf{N} = n_{1:J+1}$ of the musical notes, and

the spectrogram $\mathbf{X}$ as

$$p(\mathbf{X}, \mathbf{P}, \mathbf{N}, \mathbf{S}) = p(\mathbf{X}|\mathbf{P}, \mathbf{N})p(\mathbf{P}, \mathbf{N}, \mathbf{S}). \tag{4.1}$$

Here, all the probabilities are implicitly dependent on the tatum information $t_{1:N+1}$ and $l_{1:N+1}$. $p(\mathbf{P}, \mathbf{N}, \mathbf{S})$ represents a *language model* that describes the generative process of the musical notes and keys. $p(\mathbf{X}|\mathbf{P}, \mathbf{N})$ represents an *acoustic model* that describes the generative process of the spectrogram given the musical notes.

Given the generative model, the transcription problem can be formulated as a statistical inference problem of estimating the musical scores $(\mathbf{P}, \mathbf{N})$ and the keys $\mathbf{S}$ that maximize the left-hand side of Eq. (4.1) for the given spectrogram $\mathbf{X}$ (as explained later). In this step, the acoustic model evaluates the fitness of a musical score to the spectrogram while the language model evaluates the prior probability of the musical score. The proposed method is therefore consistent with our intuition that both of these viewpoints are essential for transcription.

4.2.3 Language Model

We construct a generative model where the pitches $\mathbf{P} = p_{1:J}$ and the onset times $\mathbf{N} = n_{1:J+1}$ are independently generated and the pitches are generated depending on the local keys $\mathbf{S} = s_{1:M}$. The generative process can be mathematically expressed as

$$p(\mathbf{P}, \mathbf{N}, \mathbf{S}) = p(\mathbf{P}|\mathbf{S})p(\mathbf{N})p(\mathbf{S}), \tag{4.2}$$

where $p(\mathbf{P}|\mathbf{S})$, $p(\mathbf{N})$, and $p(\mathbf{S})$ represent the pitch transition model, the onset time transition model, and the key transition model, respectively.

In the key transition model, to represent the sequential dependency between the keys of consecutive measures, the keys $\mathbf{S} = s_{1:M}$ are generated by a Markov model as

$$p(\mathbf{S}) = p(s_1) \prod_{m=2}^{M} p(s_m|s_{m-1}). \tag{4.3}$$

The initial and transition probabilties are parameterized as

$$p(s_1 = s) = \pi_s^{\mathrm{ini}}, \tag{4.4}$$

$$p(s_m = s \,|\, s_{m-1} = s') = \pi_{(s-s') \bmod 12 + 1}, \tag{4.5}$$

where we have assumed that the transition probabilities are symmetric under transpositions. For example, the transition probability from C major to D major is assumed to be the same as that from D major to E major. We define $\boldsymbol{\pi}^{\mathrm{ini}}=(\pi_s^{\mathrm{ini}}), \boldsymbol{\pi}=(\pi_s) \in \mathbb{R}_{\geq 0}^{12}$.

In the pitch transition model, to represent the sequential dependency of adjacent pitches and the dependency of pitches on the local keys, the pitches $\mathbf{P} = p_{1:J+1}$ are generated by a Markov model conditioned on keys $\mathbf{S} = s_{1:M}$ as

$$p(\mathbf{P}|\mathbf{S}) = p(p_1|s_1) \prod_{j=2}^{J} p(p_j|p_{j-1}, s_{m(j)}), \tag{4.6}$$

where $m(j)$ indicates the measure to which the j th note onset belongs. The initial and transition probabilties are parameterized as

$$p(p_1 = p \,|\, s_1 = s) = \phi_{sp}^{\mathrm{ini}}, \tag{4.7}$$

$$p(p_j = p \,|\, p_{j-1} = p', s_{m(j)} = s) = \phi_{sp'p}. \tag{4.8}$$

We assume that these probabilities are symmetric under transpositions so that the following relations hold:

$$\phi_{sp}^{\mathrm{ini}} \propto \bar{\phi}_{\deg(s,p)}^{\mathrm{ini}}, \tag{4.9}$$

$$\phi_{spp'} \propto \bar{\phi}_{\deg(s,p)\deg(s,p')}, \tag{4.10}$$

where

$$\deg(s,p) = \begin{cases} (p-s) \bmod 12 + 1 & (p > 0), \\ 0 & (p = 0) \end{cases} \tag{4.11}$$

represents the degree of pitch p in key s. We define $\bar{\boldsymbol{\phi}}^{\mathrm{ini}} = (\bar{\phi}_d^{\mathrm{ini}}) \in \mathbb{R}_{\geq 0}^{13}$ and $\bar{\boldsymbol{\phi}} = (\bar{\phi}_{dd'}) \in \mathbb{R}_{\geq 0}^{13 \times 13}$.

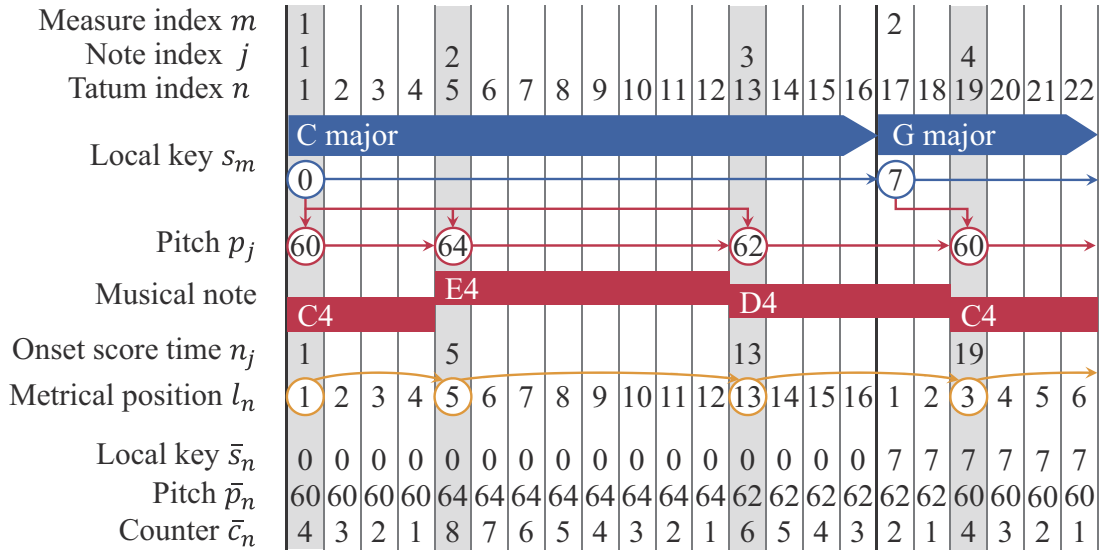


Figure 4.3: Representation of a melody note sequence and variables of the language model.

In the onset time transition model, to represent the rhythmic patterns of musical notes, the onset times $\mathbf{N} = n_{1:J+1}$ are generated by the metrical Markov model [47,48] as

$$p(\mathbf{N}) = p(n_1) \prod_{j=2}^{J+1} p(n_j | n_{j-1}), \tag{4.12}$$

where the initial and transition probabilities are given by

$$p(n_1) = \delta_{1,n_1}, \tag{4.13}$$

$$p(n_j = n \,|\, n_{j-1} = n') = \psi_{l_{n'} l_n}. \tag{4.14}$$

Here, δ denotes the Kronecker's symbol and the first equation expresses the assumption $n_1 = 1$. In the second equation, $\boldsymbol{\psi} = (\psi_{l'l}) \in \mathbb{R}_{\geq 0}^{L \times L}$ represents the transition probabilities between metrical positions.

4.2.4 Tatum-Level Language Model Formulation

In the language model presented in Section 4.2.3, the transitions of keys and transitions of pitches and onset times are not synchronized. To enable the integration with the acoustic model and the inference for AST, we here formulate

an equivalent language model where the variables are defined at the tatum level. For this purpose, we introduce tatum-level key variables $\bar{s}_n$, pitch variables $\bar{p}_n$, and counter variables $\bar{c}_n$ (Fig. 4.3). The first two sets of variables are constructed from the keys $s_{1:M}$ and the pitches $p_{1:J}$ so that $\bar{s}_n = s_m$ when tatum n is in measure m and $\bar{p}_n = p_j$ when tatum n satisfies $n_j \leq n < n_{j+1}$. The counter variable $\bar{c}_n$ represents the residual duration of the current musical note in tatum units and takes a value in $\{1, \ldots, 2L\}$, where $2L$ is the maximum length of a musical note. This variable is gradually decremented tatum by tatum until the next note begins; a note onset at tatum n is indicated by $\bar{c}_{n-1} = 1$. In this way, we can construct variables $\overline{\mathbf{S}} = \bar{s}_{1:N}$, $\overline{\mathbf{P}} = \bar{p}_{1:N}$, and $\overline{\mathbf{C}} = \bar{c}_{1:N}$ from variables $\mathbf{S} = s_{1:M}$, $\mathbf{P} = p_{1:J}$, and $\mathbf{N} = n_{1:J+1}$, and vice versa.

The generative models for the tatum-level keys, pitches, and counters can be derived from the language model in Section 4.2.3 as follows. The keys $\overline{\mathbf{S}} = \bar{s}_{1:N}$ obey the following Markov model:

$$p(\bar{\mathbf{S}}) = p(\bar{s}_1) \prod_{n=2}^{N} p(\bar{s}_n | \bar{s}_{n-1}), \tag{4.15}$$

where

$$p(\bar{s}_1) = \pi^{\mathrm{ini}}_{\bar{s}_1}, \tag{4.16}$$

$$p(\bar{s}_n | \bar{s}_{n-1}) = \begin{cases} \pi_{(\bar{s}_n - \bar{s}_{n-1}) \bmod 12 + 1} & (l_n = 1), \\ \delta_{\bar{s}_{n-1}, \bar{s}_n} & (l_n > 1). \end{cases} \tag{4.17}$$

The second equation says that a key transition occurs only at the beginning of a measure.

The counters $\overline{\mathbf{C}} = \bar{c}_{1:N}$ obey the following Markov model:

$$p(\overline{\mathbf{C}}) = p(\bar{c}_1) \prod_{n=2}^{N} p(\bar{c}_n | \bar{c}_{n-1}), \tag{4.18}$$

where

$$p(\bar{c}_1 = \bar{c}) = \psi_{l_1 l_{1+\bar{c}}}, \tag{4.19}$$

$$p(\bar{c}_n = \bar{c} \,|\, \bar{c}_{n-1} = \bar{c}') = \begin{cases} \psi_{l_n l_{n+\bar{c}}} & (\bar{c}' = 1), \\ \delta_{\bar{c}'-1, \bar{c}} & (\bar{c}' > 1). \end{cases} \tag{4.20}$$

This is a type of semi-Markov models called the residential-time Markov model [81].

The pitches $\overline{\mathbf{P}} = \bar{p}_{1:N}$ obey the following Markov model conditioned on the keys and counters:

$$p(\overline{\mathbf{P}}|\overline{\mathbf{S}}, \overline{\mathbf{C}}) = p(\bar{p}_1|\bar{s}_1) \prod_{n=2}^{N} p(\bar{p}_n|\bar{p}_{n-1}, \bar{c}_{n-1}, \bar{s}_n), \tag{4.21}$$

where

$$p(\bar{p}_1|\bar{s}_1) = \phi^{\mathrm{ini}}_{\bar{s}_1\bar{p}_1}, \tag{4.22}$$

$$p(\bar{p}_n|\bar{p}_{n-1}, \bar{c}_{n-1}, \bar{s}_n) = \begin{cases} \phi_{\bar{s}_n\bar{p}_{n-1}\bar{p}_n} & (\bar{c}_{n-1} = 1), \\ \delta_{\bar{p}_{n-1}\bar{p}_n} & (\bar{c}_{n-1} > 1). \end{cases} \tag{4.23}$$

The second equation expresses the constraint that a pitch transition occurs only at a note onset.

Putting Eqs. (4.15), (4.18), and ((4.21) together, we have

$$p(\mathbf{P}, \mathbf{N}, \mathbf{S}) = p(\overline{\mathbf{P}}, \overline{\mathbf{C}}, \overline{\mathbf{S}}) = p(\overline{\mathbf{P}}|\overline{\mathbf{S}}, \overline{\mathbf{C}})p(\overline{\mathbf{C}})p(\overline{\mathbf{S}}). \tag{4.24}$$

That is, the language model in Section 4.2.3 and the tatum-level language model defined here are equivalent probabilistic models. We use this tatum-level semi-Markov model in what follows.

4.2.5 Acoustic Model

We formulate an acoustic model $p(\mathbf{X}|\mathbf{P}, \mathbf{N}) = p(\mathbf{X}|\overline{\mathbf{P}}, \overline{\mathbf{C}})$ that gives the probability of spectrogram $\mathbf{X}$ given a pitch sequence $\overline{\mathbf{P}}$ and a counter sequence $\overline{\mathbf{C}}$ representing onset times. We define the tatum-level spectra $\mathbf{X}_n$ as a segment of spectrogram $\mathbf{X}$ in the span of tatum n. As in the standard HMM, we assume the conditional independence of the probabilities of tatum-level spectra as

$$p(\mathbf{X}|\overline{\mathbf{P}}, \overline{\mathbf{C}}) = \prod_{n=1}^{N} p(\mathbf{X}_n|\bar{p}_n, \bar{c}_{n-1}). \tag{4.25}$$

Using Bayes' theorem, the individual factors in the right-hand side of Eq. (4.25) can be written as

$$p(\mathbf{X}_n|\bar{p}_n, \bar{c}_{n-1}) = \frac{p(\bar{p}_n, \bar{c}_{n-1}|\mathbf{X}_n)p(\mathbf{X}_n)}{p(\bar{p}_n, \bar{c}_{n-1})} \tag{4.26}$$

$$\propto \frac{p(\bar{p}_n, \bar{c}_{n-1}|\mathbf{X}_n)}{p(\bar{p}_n, \bar{c}_{n-1})}, \tag{4.27}$$

where $p(\bar{p}_n, \bar{c}_{n-1})$ is the *prior* probability of pitch $\bar{p}_n$ and counter $\bar{c}_{n-1}$ and $p(\bar{p}_n, \bar{c}_{n-1}|\mathbf{X}_n)$ is the *posterior* probability of $\bar{p}_n$ and $\bar{c}_{n-1}$.

We use a CRNN for estimating the probability $p(\bar{p}_n, \bar{c}_{n-1}|\mathbf{X}_n)$ (Fig. 4.4). Since it is considered to be difficult to directly estimate the counter variables describing the durations of musical notes from the locally observed quantity $\mathbf{X}_n$, we train the CRNN to predict the probability whether a note onset occurs at each tatum. A similar DNN for joint estimation of pitch and onset probabilities has been successfully applied to piano transcription [41]. For reliable estimation, we estimate the pitch and onset probabilities independently. Therefore, the CRNN takes the spectra $\mathbf{X}_n$ as input and outputs the following probabilities:

$$\xi_{nk} = p(\bar{p}_n = k|\mathbf{X}_n), \tag{4.28}$$

$$\zeta_n = p(\bar{o}_n = 1|\mathbf{X}_n), \tag{4.29}$$

where $\bar{o}_n \in \{0, 1\}$ is an onset flag and ζ_n is the (posterior) onset probability at tatum n ($\bar{o}_n = 1$ if there is a note onset at tatum n and $\bar{o}_n = 0$ otherwise). The counter probabilities are assigned using the onset probability as

$$p(\bar{c}_{n-1}|\mathbf{X}_n) = \begin{cases} \zeta_n & (\bar{c}_{n-1} = 1), \\ (1 - \zeta_n)/(L - 1) & (\bar{c}_{n-1} \neq 1), \end{cases} \tag{4.30}$$

and the probability $p(\bar{p}_n, \bar{c}_{n-1}|\mathbf{X}_n)$ is then given as the product of Eqs. (4.28) and (4.30).

In practice, it is reasonable to use spectra of a longer segment than a tatum as input of the CRNN since each tatum (16th note) spans a short time interval. In addition, it is computationally efficient to jointly estimate the pitch and onset probabilities of all tatums in the wide segment. Therefore, we use the whole spectrogram $\mathbf{X}$ (or its part of a sufficient duration) as the input of the CRNN and train it so as to output all the tatum-level pitch and onset probabilities.

The CRNN consists of a frame-level CNN and a tatum-level RNN linked through a max-pooling layer (Fig. 4.4). The CNN extracts latent features $\mathbf{e}_{1:T}$

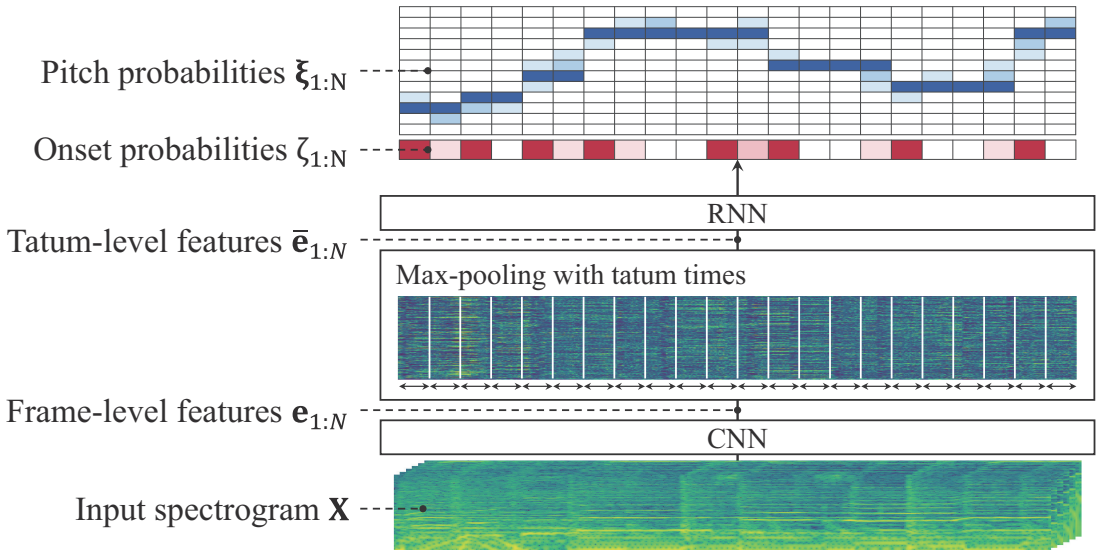


Figure 4.4: The acoustic model $p(\mathbf{X}|\overline{\mathbf{P}}, \overline{\mathbf{C}})$ representing the generative process of the spectrogram $\mathbf{X}$ from note pitches $\overline{\mathbf{P}}$ and residual durations $\overline{\mathbf{C}}$.

($\mathbf{e}_t = [e_{t1}, \ldots, e_{tF}] \in \mathbb{R}^F$) from the spectrogram $\mathbf{X}$ of length T:

$$\mathbf{e}_{1:T} = \mathrm{CNN}(\mathbf{X}). \tag{4.31}$$

Using the tatum times $t_{1:N+1}$, the max-pooling layer summarizes the frame-level features $\mathbf{e}_{1:T}$ into the tatum-level features $\bar{\mathbf{e}}_{1:N}$ ($\bar{\mathbf{e}}_n = [\bar{e}_{n1}, \ldots, \bar{e}_{nF}] \in \mathbb{R}^F$) as

$$\bar{e}_{nf} = \max_{t_n \leq t < t_{n+1}} e_{tf}. \tag{4.32}$$

The RNN then converts the tatum-level features $\bar{\mathbf{e}}_{1:N}$ into intermediate features $\mathbf{g}_{1:N}$ ($\mathbf{g}_n \in \mathbb{R}^D$ is a D-dimensional vector) through a bidirectional long short-term memory (BLSTM) layer and predicts the pitch and onset probabilities $\boldsymbol{\xi}_n = (\xi_{nk}) \in \mathbb{R}^{K+1}$ and ζ_n through softmax and sigmoid layers as follows:

$$\mathbf{g}_{1:N} = \mathrm{BLSTM}(\bar{\mathbf{e}}_{1:N}), \tag{4.33}$$

$$\boldsymbol{\xi}_n = \mathrm{Softmax}\left(\mathbf{W}^{\mathrm{p}}\mathbf{g}_n + \mathbf{b}^{\mathrm{p}}\right), \tag{4.34}$$

$$\zeta_n = \mathrm{Sigmoid}\left(\mathbf{W}^{\mathrm{o}}\mathbf{g}_n + b^{\mathrm{o}}\right), \tag{4.35}$$

where $\mathbf{W}^{\mathrm{p}} \in \mathbb{R}^{(K+1)\times D}$ and $\mathbf{W}^{\mathrm{o}} \in \mathbb{R}^{1\times D}$ are weight matrices, and $\mathbf{b}^{\mathrm{p}} \in \mathbb{R}^{K+1}$ and $b^{\mathrm{o}} \in \mathbb{R}$ are bias parameters.

4.2.6 Training Model Parameters

The parameters $\boldsymbol{\psi}$, $\bar{\boldsymbol{\phi}}^{\mathrm{ini}}$, $\bar{\boldsymbol{\phi}}$, $\boldsymbol{\pi}^{\mathrm{ini}}$, and $\boldsymbol{\pi}$ of the language model are learned from training data of musical scores. The metrical transition probabilities $\psi_{ll'}$ are estimated as

$$\psi_{ll'} \propto \max(a_{ll'} - \kappa, 0) + \epsilon, \tag{4.36}$$

where $a_{ll'}$ is the number of transitions from metrical positions l to l' appear in the training data, κ is a discount parameter, and ϵ is a small value to avoid the zero count. The initial and transition probabilities of pitches ($\bar{\boldsymbol{\phi}}^{\mathrm{ini}}$ and $\bar{\boldsymbol{\phi}}$) are estimated in the same way, by using the key signatures. Although the initial and transition probabilities of local keys ($\boldsymbol{\pi}^{\mathrm{ini}}$ and $\boldsymbol{\pi}$) can be trained in the same way in principle, a large amount of musical scores are necessary for reliable estimation since modulations are rare. Therefore, in this study, we manually set $\boldsymbol{\pi}^{\mathrm{ini}}$ to the uniform distribution and $\boldsymbol{\pi}$ to $[0.9, 0.1/11, \cdots, 0.1/11]$ such that the self-transition probability π_0 has a large value.

The parameters of the CRNN are trained by using paired data of audio spectrograms and corresponding musical scores. After converting the pitches and onset times into the form $\overline{\mathbf{P}} = \bar{p}_{1:N}$ and $\overline{\mathbf{O}} = \bar{o}_{1:N}$, we apply the following cross-entropy loss functions to train the CRNN:

$$\mathcal{L}_{\mathrm{pitch}} = -\sum_{n=1}^{N}\sum_{k=0}^{K} \delta_{\bar{p}_n,k} \log \xi_{nk}, \tag{4.37}$$

$$\mathcal{L}_{\mathrm{onset}} = -\sum_{n=1}^{N} \big\{ \bar{o}_n \log \zeta_n + (1-\bar{o}_n) \log (1-\zeta_n) \big\}, \tag{4.38}$$

4.2.7 Transcription Algorithm

We can derive a transcription algorithm based on the constructed generative model. Using the tatum-level formulation, Eq. (4.1) can be rewritten as

$$p(\mathbf{X}, \overline{\mathbf{S}}, \overline{\mathbf{P}}, \overline{\mathbf{C}}) = p(\mathbf{X}|\overline{\mathbf{P}}, \overline{\mathbf{C}}) p(\overline{\mathbf{P}}, \overline{\mathbf{C}}, \overline{\mathbf{S}}), \tag{4.39}$$

where the first factor on the right-hand side is given by the CRNN as in Eq. (4.25) and the second factor by the semi-Markov model as in Eq. (4.24). Therefore,

the integrated generative model is a CRNN-HSMM hybrid model. The most likely musical score can be estimated from the observed spectrogram $\mathbf{X}$ by maximizing the probability $p(\overline{\mathbf{S}}, \overline{\mathbf{P}}, \overline{\mathbf{C}}|\mathbf{X}) \propto p(\mathbf{X}, \overline{\mathbf{S}}, \overline{\mathbf{P}}, \overline{\mathbf{C}})$, where we have used Bayes' formula. In equation, we estimate the optimal keys $\overline{\mathbf{S}}^* = \bar{s}^*_{1:N}$, pitches $\overline{\mathbf{P}}^* = \bar{p}^*_{1:N}$, and counters $\overline{\mathbf{C}}^* = \bar{c}^*_{1:N}$ at the tatum level such that

$$\overline{\mathbf{S}}^*, \overline{\mathbf{P}}^*, \overline{\mathbf{C}}^* = \operatorname{argmax} p(\mathbf{X}, \overline{\mathbf{S}}, \overline{\mathbf{P}}, \overline{\mathbf{C}}). \tag{4.40}$$

The most likely pitches $\mathbf{P}^* = p^*_{1:J}$ and onset times $\mathbf{N}^* = n^*_{1:J}$ of musical notes can be obtained from $\overline{\mathbf{P}}^*$ and $\overline{\mathbf{C}}^*$. The number of notes J is determined in this inference process.

Viterbi Algorithm

We can use the Viterbi algorithm to solve Eq. (4.40). In the forward step, viterbi variables $\omega_n(q_n)$, where $q_n = \{\bar{s}_n, \bar{p}_n, \bar{c}_n\}$, are recursively calculated as follows:

$$\begin{aligned}\omega_1(q_1) = {} & \frac{p(\bar{p}_1|\mathbf{X}_1)^{\beta_\xi} p(\bar{c}_0{=}1|\mathbf{X}_1)^{\beta_\zeta}}{p(\bar{p}_1, \bar{c}_0{=}1)^{\beta_\chi}} \\ & \cdot p(\bar{s}_1)^{\beta_\pi} p(c_1)^{\beta_\psi} p(\bar{p}_1|\bar{s}_1)^{\beta_\phi}, \end{aligned} \tag{4.41}$$

$$\begin{aligned}\omega_n(q_n) = \max_{q_{n-1}} \Bigg(& \frac{p(\bar{p}_n|\mathbf{X}_n)^{\beta_\xi} p(\bar{c}_{n-1}|\mathbf{X}_n)^{\beta_\zeta}}{p(\bar{p}_n, \bar{c}_{n-1})^{\beta_\chi}} \\ & \cdot p(\bar{s}_n|\bar{s}_{n-1})^{\beta_\pi} p(\bar{c}_n|\bar{c}_{n-1})^{\beta_\psi} \\ & \cdot p(\bar{p}_n|\bar{p}_{n-1}, \bar{c}_{n-1}, \bar{s}_n)^{\beta_\phi} \Bigg), \end{aligned} \tag{4.42}$$

where $\bar{c}_0 = 1$ was formally introduced in the initialization. In the above equations, we have introduced weighting factors β_π, β_ϕ, β_ψ, β_ξ, β_ζ, and β_χ to balance the language model and the acoustic model. In the recursive calculation, q_{n-1} that maximizes the max operation is memorized as $\mathrm{prev}(q_n)$.

In the backward step, the optimal variables $q^*_{1:N}$ are recursively obtained as follows:

$$q^*_N = \operatorname*{argmax}_{q_N} \omega_N(q_N), \tag{4.43}$$

$$q^*_n = \mathrm{prev}(q^*_{n+1}). \tag{4.44}$$

Refinements

Musical scores estimated by the CRNN-HSMM tend to have long durations because the accumulative multiplication of pitch and onset time transition probabilities decreases the posterior probability. This is known as a general problem of the HSMM [90]. To ameliorate this situation, we penalize long notes by multiplying each of Eqs. (4.41) and (4.42) by the following penalty term:

$$f(\bar{c}_{n-1}, \bar{c}_n) = \begin{cases} \{\exp(1/\bar{c}_n)\}^{\beta_\eta} & (\bar{c}_{n-1} = 1), \\ 1 & (\bar{c}_{n-1} \neq 1), \end{cases} \tag{4.45}$$

where $\beta_\eta \geq 0$ is a weighting factor.

To save the computational costs of the Viterbi algorithm defined in the large product space $q_n = \{\bar{s}_n, \bar{p}_n, \bar{c}_n\}$ without sacrifice of its global optimality, we limit the pitch space to be searched as follows:

$$\bar{p}_n \in \bigcup_{n'=n-1}^{n+1} \underset{0 \leq p \leq K}{\mathrm{top3}} \, (\xi_{n'p}), \tag{4.46}$$

where $\mathrm{top3}_{0 \leq p \leq K}(\xi_{np})$ represents the set of the indices p that provide the three largest elements in $\{\xi_{n0}, \ldots, \xi_{nK}\}$.

4.3 Evaluation

We report comparative experiments conducted for evaluating the proposed AST method. We compared the proposed method with existing methods and examined the effectiveness of the language model (Section 4.3.3). We then investigated the AST performance of the proposed method for music and singing signals with different complexities and examined the influence of the beat tracking performance on the AST performance (Section 4.3.4).

4.3.1 Data

We used 61 popular songs with reliable melody annotations [84] from the RWC music database [83]. We split the data into a training dataset (37 songs), a validation dataset (12 songs), and a test dataset (12 songs), where the singers of

these datasets are disjoint. We also used 20 synthesized singing signals obtained by a singing synthesis software called CeVIO [94]; 12, 4, and 4 signals are added to the training, validation, and test datasets, respectively.

To augment the training data, we added the separated singing signals obtained by Spleeter [18] and the clean isolated singing signals. To cover a wide range of pitches and tempos, we pitch-shifted the original music signals by L semitones ($-12 \leq L \leq 12$) and randomly time-stretched each of those signals with a ratio of R ($0.5 \leq R < 1.5$). The total number of songs in the training data was $37 \times 25 \times 3$ (real) $+ 49 \times 25$ (synthetic) $= 4000$.

For each signal sampled at 22.05 kHz, we used a short-time Fourier transform (STFT) with a Hann window of 2048 points and a shifting interval of 256 points (11.6 ms) for calculating the amplitude spectrogram on the logarithmic frequency axis having 5 bins per semitone (*i.e.*, 1 bin per 20 cents) between 32.7 Hz (C1) and 2093 Hz (C7) [95]. We then computed the HCQT-like spectrogram $\mathbf{X}$ by stacking the h-harmonic-shifted versions of the original spectrogram, where $h \in \{1/2, 1, 2, 3, 4, 5\}$ (*i.e.*, $H = 6$), and the lowest and highest frequencies of the h-harmonic-shifted spectrogram are $h \times 32.7$ Hz and $h \times 2093$ Hz, respectively.

4.3.2 Setup

Inspired by the CNN proposed for frame-level melody F0 estimation [9], the frame-level CNN of the acoustic model (Fig. 4.5) was designed to have six convolution layers with the output channels of 128, 64, 64, 64, 8, and 1 and the kernel sizes of $(5, 5)$, $(5, 5)$, $(3, 3)$, $(3, 3)$, $(70, 3)$, and $(1, 1)$, respectively, where the instance normalization [96] and the Mish function [97] are used. The output dimension of the tatum-level BLSTM was set to $D = 130 \times 2$. The vocabulary of pitches consisted of a rest and 128 semitone-level pitches specified by the MIDI note numbers ($K = 128$).

To optimize the proposed CRNN, we used RAdam with the parameters $\alpha = 0.001$ (learning rate), $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. A weight decay (L2 regularization) with a hyperparameter 10^{-5} and a gradient clipping with a threshold of 5.0 were used for training. The weight parameters $\mathbf{W}^{\mathrm{p}}$ and $\mathbf{W}^{\mathrm{o}}$

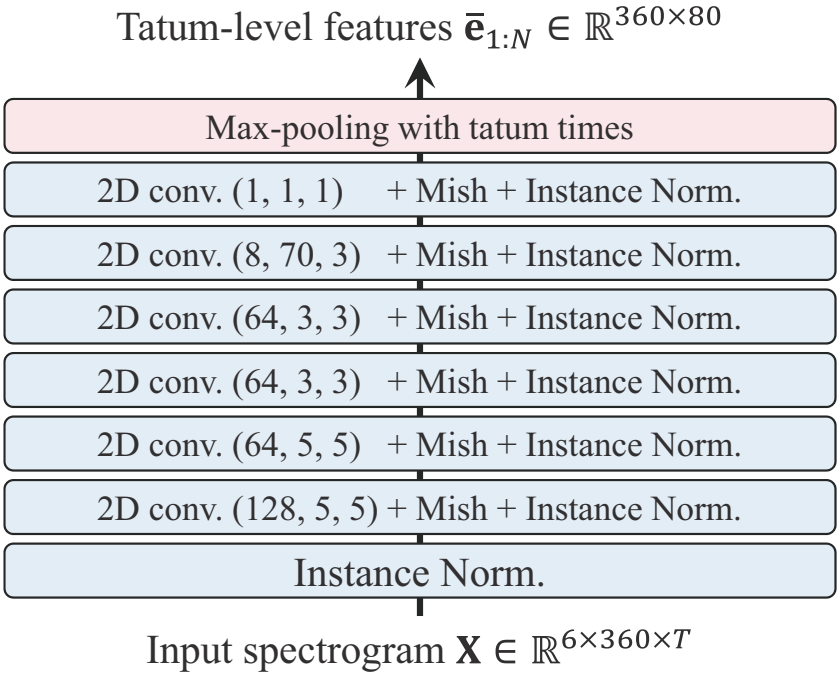


Figure 4.5: Architecture of the CNN. Three numbers in the parentheses in each layer indicate the channel size, height, and width of the kernel.

were initialized to random values between -0.1 and 0.1. The kernel filters of the frame-level CNN and the weight parameters of the tatum-level BLSTM were initialized by He's method [98]. All bias parameters were initialized with zero.

Because of the limited memory capacity, we split the spectrogram of each song into 80-tatums segments. The CRNN's outputs of those segments were concatenated for the note estimation based on the Viterbi decoding. The weighting factors β_π, β_ϕ, β_ψ, β_ξ, β_ζ, and β_η were optimized for the validation data by using Optuna [99]. Consequently, $\beta_\pi = 0.541$, $\beta_\phi = 0.769$, $\beta_\psi = 0.619$, $\beta_\xi = 0.917$, $\beta_\zeta = 0.852$, and $\beta_\eta = 0.609$. We manually set the weighting factor β_χ to 0 based on the results of preliminary experiments. The discounting value κ and small value ϵ in Eq. (4.36) were set to 0.7 and 0.1, respectively.

The accuracy of estimated musical notes was measured with the edit-distance-based metrics proposed in [50] consisting of pitch error rate E_p, missing note rate E_m, extra note rate E_e, onset error rate E_on, offset error rate E_off, and overall (average) error rate E_all.

Table 4.1: The AST performances (%) of the different methods.

Method	Signal	F0	Tatums	E_{p}	E_{m}	E_{e}	E_{on}	E_{off}	E_{all}
CRNN-HSMM (proposed)	mix.	–	[24]	8.34	13.50	13.70	24.45	23.06	16.61
	sep. [18]			7.85	9.63	14.45	22.46	21.64	15.21
CRNN	mix.	–	[24]	7.81	12.07	18.00	29.35	28.05	19.06
	sep. [18]			8.56	8.60	17.57	31.12	27.21	18.61
HSMM	mix.	[11]	[24]	8.74	33.07	16.96	53.02	33.29	29.02
	sep. [18]			9.81	31.80	15.79	52.68	31.79	28.38
Majority vote	mix.	[11]	[24]	20.52	7.02	32.23	58.46	49.36	33.52
	sep. [18]			20.55	7.69	32.90	59.38	50.63	34.23

4.3.3 Method Comparison

To confirm the AST performance of the proposed method, we compared the transcription results obtained by the proposed CRNN-HSMM hybrid model, the HSMM-based method [90], and the majority-vote method. The majority-vote method quantizes an input F0 trajectory in semitone units, then determines tatum-level pitches by taking the majority of the quantized pitches at each tatum. Since the majority-vote method does not estimate note onsets, we concatenated successive tatums with the same pitch to obtain a single musical note. For the HSMM-based method and the majority-vote method, the F0 contours estimated by [11] were used as inputs.

To examine the effect of the language model, we also run a method using only the CRNN as follows:

$$p_n^* = \underset{1 \leq p \leq K}{\operatorname{argmax}} \, \xi_{np}, \tag{4.47}$$

$$o_n^* = \begin{cases} 0 & (\zeta_n < 0.5), \\ 1 & (\zeta_n \geq 0.5). \end{cases} \tag{4.48}$$

To construct a musical score from the predicted symbols p_n^* and o_n^*, we applied the following rules:

1. If $p_{n-1}^* \neq p_n^*$, then the $(n-1)$ th and n th tatums are included in different notes.

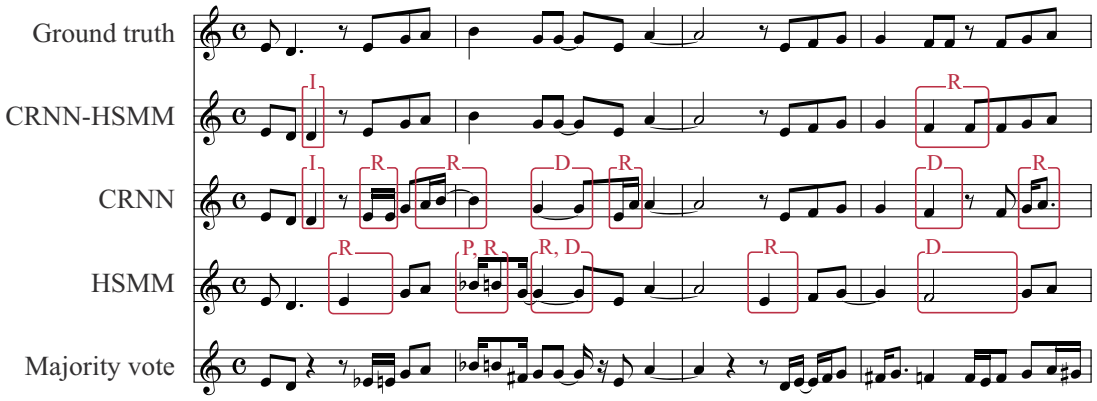


Figure 4.6: Examples of musical scores estimated by the proposed method, the CRNN method, the HSMM-based method, and the majority-vote method from the mixture and separated audio signals and the estimated F0 contours and tatum times. Transcription errors are indicated by the red squares. Capital letters attached to the red squares represent the following error types: pitch error (P), rhythm error (R), deletion error (D), and insertion error (I). Error labels are not shown in the transcription result by the majority-vote method, which contains too many errors.

2. If $p^*_{n-1} = p^*_n$ and $o^*_n = 1$, then the $(n-1)$ th and n th tatums are included in different notes having the same pitch.
3. If $p^*_{n-1} = p^*_n$ and $o^*_n = 0$, then the $(n-1)$ th and n th tatums are included in the same notes.

To evaluate the methods in a realistic situation, only the mixture signals and the separated signals were used as test data, and the tatum times were estimated by [24].

Results are shown in Table 4.1. For both the mixture and separated signals, the proposed method and the CRNN method outperformed the majority-vote method and the HSMM-based method in the overall error rate E_{all} by large margins. This result confirms the effectiveness of using the CRNN as the acoustic model. Comparing the E_{all} metrics for the proposed method and the CRNN method, there was a decrease of 2.33 percentage points (PP) for the mixture signals and 1.62 PP for the separated signals. This result indicates the positive effect of the language model. Especially, the significant decreases of the E_{on} and E_{off} metrics indicate that the language model is particularly effective for reducing

Table 4.2: The AST performances (%) obtained from the different input data.

Method	Signal	Tatums	E_{p}	E_{m}	E_{e}	E_{on}	E_{off}	E_{all}
CRNN-HSMM	mix.	ground-truth	7.30	13.81	14.76	24.46	22.80	16.62
	sep. [18]		7.89	8.42	13.18	21.81	20.24	14.31
	iso.		6.68	6.64	8.56	16.81	16.47	11.03
	syn.		0.00	0.10	0.47	0.34	1.69	0.52
	mix.	[24]	8.34	13.50	13.70	24.45	23.06	16.61
	sep. [18]		7.85	9.63	14.45	22.46	21.64	15.21
	iso.		6.83	6.55	10.31	19.17	18.79	12.33
	syn.		0.42	2.87	1.52	11.25	6.97	4.60

rhythm errors. The proposed method and the CRNN method achieved better performances for the separated signals than the mixture signals.

Transcription examples obtained by the different models are shown in Fig. 4.6 [1]. The musical score estimated by the majority-vote method, which did not use a language model, had many errors. In the musical score estimated by the HSMM-based method, whereas most notes had pitches on the musical scale, repeated note onsets with the same pitches were not detected. In the result by the CRNN method, which did not use a language model, we can see that most pitches are on the musical scale unlike in the result by the majority-vote method. This indicates the capacity of the CRNN that some sequential constraints on musical notes can be learned by the RNN. However, there were some errors in rhythms, which suggests the difficulty of learning rhythmic constraints by a simple RNN. Finally, in the result by the proposed CRNN-HSMM method, there were much fewer rhythm errors than the CRNN method, which demonstrates the effect of the language model.

4.3.4 Influences of Voice Separation and Beat Tracking Methods

A voice separation method [18] and a beat-tracking method [24] are used in the preprocessing step of the proposed method, and errors made in this step can propagate to the final transcription results. Here, we investigate the influences

[1] Other examples are available in the accompanying webpage: `http://sap.ist.i.kyoto-u.ac.jp/members/nishikimi/demo/apsipa-tsip-2020/`

of those methods used in the preprocessing step. We used the ground-truth tatum times obtained from the beat annotations [84] to examine the influence of the beat-tracking method. We used the isolated signals for the songs in the test data to examine the influence of the voice separation method. In addition, as a reference, we also evaluated the proposed method with the synthetic singing voices. When tatum times were estimated by the beat-tracking method [24] for the real signals, the mixture signals are used as input and the results are used for the mixture, separated, and isolated signals. Since the synthesized signals are not synchronized to the mixture signals, the beat-tracking method is directly applied to the vocal signals to obtain estimated tatum times.

Results are shown in Table 4.2. As for the influence of the beat-tracking method, using the ground-truth tatum times decreased the overall error rate E_{all} by 1.1 PP for the separated signals and 1.3 PP for the isolated signals. This result indicates that the influence of the beat-tracking method is small for the data used. As for the influence of the voice separation method, in both conditions with estimated and ground-truth tatum times, E_{all} for the isolated signals were approximately 3 PP smaller than that for the separated signals. This result indicates that further refinements on the voice separation method can improve the transcription results by the proposed method. Finally, in both conditions with estimated and ground-truth tatum times, the transcription error rates for the synthesized signals were significantly smaller than those for the real signals. This result confirms that the difficulty of AST originates from the pitch and timing deviations in sung melodies. The relatively large onset and offset error rates for the case of using the estimated tatum times are due to the difficulty of beat tracking for the signals containing only a singing voice.

4.3.5 Discussion

Our results provide an important insight that a simple RNN has a weak effect in capturing the rhythmic structure and the language model that explicitly incorporates a rhythm model plays a significant role in improving the transcription results. Whereas musical pitches can be inferred from local acoustic features,

in order to recognize musical rhythms, it is necessary to look at durations or intervals of onset times, which have extended structures in time. This non-local feature of rhythms characterizes the difficulty of music transcription, which cannot be solved by simply applying DNN techniques used for other tasks such as automatic speech recognition (ASR). This result may also explain why end-to-end methods that were successful at ASR have not been so successful at music transcription [53, 56, 91, 92]. For example, the paper [56] reports low error rates for monophonic transcription, but the method was only applied to synthetic data without timing deviations.

To simplify the AST task, we imposed the following restrictions on target songs in this study: the tatum unit (minimal resolution of a beat) is a 16th-note length and the meter of a target song is $4/4$ time. Theoretically, we can relax these restrictions by modifying the language model and extend the present method for more general target songs. To include shorter note lengths and triplets, we can introduce a shorter tatum unit, for example, a tatum corresponding to one-third of a 32nd note. To transcribe songs in other meters such as $3/4$ time, we can construct one metrical Markov model for each meter and estimate the meter of a given song by the maximum likelihood estimation [50]. Although most beat-tracking methods such as [24] assumes a constant meter for each song, popular music songs often have mixed meters (*e.g.*, an insertion of a measure in $2/4$ time), which calls for a more general rhythm model. A possible solution is to introduce latent variables representing meters (one for each measure) into the language model and estimate the variables in the transcription step.

The language model based on the first-order Markov model was used in this study and it is possible to apply more refined language models. A simple direction is to use higher-order Markov models or a neural language model based on RNN. While most language models try to capture local sequential dependence of symbols, using a model incorporating a global repetitive structure is effective for music transcription. To incorporate the repetitive structure in a computationally tractable way, it is considered to be effective to use a Bayesian language model [100].

Another important direction for refining the method would be to integrate the voice separation and/or the beat tracking with the musical note estimation method. A voice separation method and a beat-tracking method are used in the preprocessing step in the present method, and we observed that errors made in the preprocessing step can propagate to the transcription results. To mitigate the problem, multi-task learning of the singing voice separation and the AST can also be effective in obtaining the singing voices appropriate for the AST [11]. A beat-tracking method typically estimates beat times in the accompaniment sounds, which can be slightly shifted from the onset times of the singing voice due to the asynchrony between the vocal and the other parts [78]. Therefore, it would be effective for AST to jointly estimate musical notes and tatum times that match the onset times of singing voices.

4.4 Summary

This chapter presented an audio-to-score AST method based on a CRNN-HSMM hybrid model that integrates a language model with a DNN-based acoustic model. The proposed method outperformed the majority-vote method and the previously state-of-the-art HSMM-based method. We also found that the language model has a positive effect on improving the AST performance, especially in rhythmic aspects.

Chapter 5

Discriminative Approach Based on Encoder-Decoder Model with Note-Level Output

This chapter presents the discriminative approach to AST based on the encoder-decoder model with the note-level decoder. This approach attempts to avoid the error propagation caused by estimating F0 trajectories and tatum times required by the generative and hybrid approaches. In addition, we propose the semi-supervised learning framework for the attention mechanism to overcome the weakness of the encoder-decoder model in predicting the temporal attributes.

5.1 Introduction

Many studies have been conducted on AST. A naive approach to AST is to sequentially use a singing voice separation method that extracts singing voice from music audio signals [8, 13, 14] and an F0 estimation method that estimates F0 trajectories from singing voice [3, 5–8, 38, 101]. This approach requires an additional step to estimate the semitone-level pitches and note values of musical notes by quantizing F0 trajectories in the time and frequency domains. Most studies, however, have focused on only pitch quantization for estimating piano rolls [29, 30, 33], while note-value quantization (*a.k.a.* rhythm transcription) has been investigated independently [46–48]. Some studies have tried to jointly estimate the pitches and note values of musical notes from F0 trajectories using

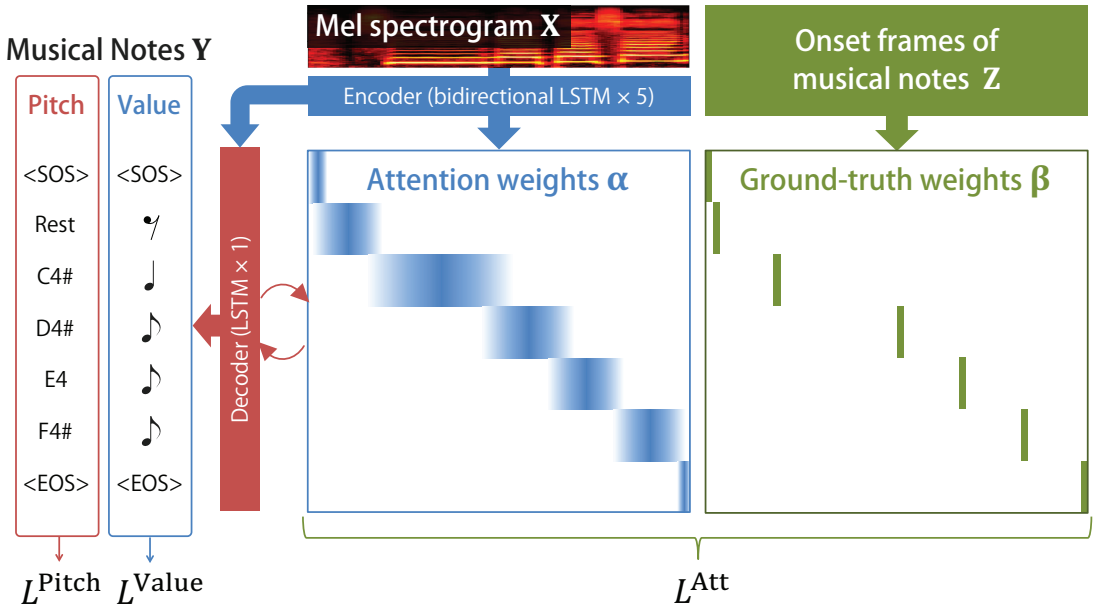


Figure 5.1: Our encoder-decoder model with an attention mechanism for end-to-end AST. This model is trained by minimizing the weighted sum of loss functions for ground-truth pitches and note values, as well as alignment information (onset times) if available.

a musical language model that represents the pitches, rhythms, and scales of musical notes [77, 78]. The major problem of the aforementioned methods are that the errors occurred in the F0 estimation adversely affect the note estimation.

In this chapter, we propose an end-to-end approach for AST. AST is similar to automatic speech recognition (ASR) in that audio spectra are converted into a sequence of meaningful symbols (musical notes or characters). Inspired by the success of sequence-to-sequence or end-to-end learning in ASR [69–71] and machine translation [66, 67], we focus on an encoder-decoder model with an attention mechanism for directly estimating a sequence of musical notes from a sequence of vocal spectra. The encoder-decoder model is a deep neural network (DNN) that can convert an input sequence into an output sequence of different lengths. The attention mechanism is an efficient method for dealing with long sequences by setting attention weights (weak alignment) between output and input sequences.

A key difference between AST and ASR is that temporal categories (note

values) must be estimated in addition to instantaneous categories (pitches). To deal with such fine temporal structure as rhythms, it is crucial to train attention weights precisely. Although a simple solution is to apply supervised learning for the attention weights, instead of the common unsupervised learning, there is a problem of the limitation of available training data for AST.

To solve this, we propose a novel framework based on semi-supervised learning for the attention weights by using a limited amount of partially inaccurate time-aligned data as guiding information (Fig. 5.1). Specifically, we propose a novel loss function that evaluates the input-output alignment estimated by the attention mechanism. If time-aligned data are available, we calculate the cross-entropy loss between the estimated normalized attention weights and the corresponding ground-truth weights for each note. We find that when the model is trained successfully in an unsupervised manner, the attention weights of each note tend to concentrate around its onset frame. As the ground-truth weights for each note, therefore, we choose a one-hot vector that is peaked at the onset frame of the note. Although the onset time annotations are often unreliable, the model can learn correct alignment by gradually reducing the weight of the attention loss function. The main contribution of this study is an easy-to-implement general technique for effectively avoiding bad local optima and accelerating convergence in the framework of semi-supervised end-to-end learning when both aligned and non-aligned data are available.

5.2 Method

This section explains the proposed method of AST with a modified attention-based encoder-decoder model (Fig. 5.1).

5.2.1 Problem Specification

Our goal is to train an encoder-decoder model that takes as input a mel-scale spectrogram $\mathbf{X} = \mathbf{x}_{1:T} \in \mathbb{R}^{F \times T}$ of an isolated solo singing voice and outputs a sequence of musical notes $\mathbf{Y} = y_{1:N} = (p_n, v_n)_{1:N}$ by using the onset frames

of those notes $\mathbf{Z} = z_{1:N}$ if available, where T, F, and N indicate the number of time frames, that of frequency bins, and that of musical notes, respectively. $\mathbf{x}_t \in \mathbb{R}^F$ indicates the mel-scale spectrum at frame t. Each note y_n is represented as a pair of a semitone-level pitch $p_n \in \{1, \ldots, K\}$ and a 16th-note-level duration $v_n \in \{1, \ldots, L\}$. K and L indicate the size of the pitch vocabulary (including the rest) and that of the note-value vocabulary, respectively. $z_n \in \{1, \ldots, T\}$ indicates the onset frame of the musical note y_n in the spectrogram $\mathbf{X}$.

5.2.2 Pitch and Note Value Decoder

In AST, it is necessary to output two symbols: a pitch and a note value. One possibility is to regard the pair of a pitch and a note value as one symbol. Such a model, however, is difficult to train from a limited amount of training data because the joint vocabulary size is $I = K \times L$. To reduce the vocabulary size to $I = K + L$, we extend the decoder to separately output both symbols at the same time as follows:

$$\boldsymbol{\phi} = \mathrm{Softmax}\big(\hat{\mathbf{P}}\mathbf{s}_{n-1} + \hat{\mathbf{Q}}\mathbf{g}_n + \hat{\mathbf{b}}\big), \tag{5.1}$$

$$p_n = \underset{p_n \in \{1,\ldots,K\}}{\mathrm{argmax}} \left(\phi_{p_n}\right), \tag{5.2}$$

$$\boldsymbol{\psi} = \mathrm{Softmax}\big(\bar{\mathbf{P}}\mathbf{s}_{n-1} + \bar{\mathbf{Q}}\mathbf{g}_n + \bar{\mathbf{b}}\big), \tag{5.3}$$

$$v_n = \underset{v_n \in \{1,\ldots,L\}}{\mathrm{argmax}} \left(\psi_{v_n}\right), \tag{5.4}$$

where $\hat{\mathbf{P}} \in \mathbb{R}^{K \times D}$, $\hat{\mathbf{Q}} \in \mathbb{R}^{K \times E}$, $\bar{\mathbf{P}} \in \mathbb{R}^{L \times D}$, and $\bar{\mathbf{Q}} \in \mathbb{R}^{L \times E}$ indicate weight matrices, and $\hat{\mathbf{b}} \in \mathbb{R}^K$ and $\bar{\mathbf{b}} \in \mathbb{R}^L$ indicate bias parameters. To treat a pitch and a note value as separate symbols, we use two different $\langle$`sos`$\rangle$ and $\langle$`eos`$\rangle$ symbols for the pitch and note-value prediction, In short, the pitch vocabulary includes $\langle$`sos_p`$\rangle$ and $\langle$`eos_p`$\rangle$, and the note-value vocabulary includes $\langle$`sos_v`$\rangle$ and $\langle$`eos_v`$\rangle$.

5.2.3 Loss Function for Attention Weights

We define a loss $\mathcal{L}^{\mathrm{Att}}$ for attention weights $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]^\top \in \mathbb{R}^{N \times T}$ to guide them to ideal values. To calculate $\mathcal{L}^{\mathrm{Att}}$, the guiding ground-truth weights $\boldsymbol{\beta}$ for

$\boldsymbol{\alpha}$ are introduced by using the optional data $\mathbf{Z}$. We find that when the model is successfully trained without calculating $\mathcal{L}^{\mathrm{Att}}$, the attention weights of each note tend to concentrate around its onset frame. A one-hot vector is thus considered to be a good choice for $\boldsymbol{\beta}$ as follows:

$$\beta_{nt} = \begin{cases} 1 & (z_n = t) \\ \varepsilon & (\text{otherwise}) \end{cases}, \tag{5.5}$$

where ε is a small positive number. After each row of $\boldsymbol{\beta}$ is normalized, $\mathcal{L}^{\mathrm{Att}}$ is given by the cross entropy as follows:

$$\mathcal{L}^{\mathrm{Att}} = -\frac{\lambda}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \beta_{nt} \log \alpha_{nt}, \tag{5.6}$$

where λ is a hyperparameter to scale loss $\mathcal{L}^{\mathrm{Att}}$, which is gradually decreased during training to learn optimal input-output alignment that is considered to be different from hard alignment $\boldsymbol{\beta}$.

5.2.4 Training and Inference Algorithms

In the training phase, the pitch and note value are separately converted to one-hot vectors, and then input to the decoder. The loss $\mathcal{L}^{\mathrm{Pitch}}$ for the output pitches and the loss $\mathcal{L}^{\mathrm{Value}}$ for the output note values are given by their cross entropies. The sum of $\mathcal{L}^{\mathrm{Pitch}}$, $\mathcal{L}^{\mathrm{Value}}$, and $\mathcal{L}^{\mathrm{Att}}$ is defined as the total loss function to be minimized. In the inference phase, the pitch and note value obtained by Eqs. (5.2) and (5.4) at the previous time step are converted into one-hot vectors and used for predicting the current symbol. This process stops when the output sequence reaches a specified maximum length or when ⟨`eos_p`⟩ or ⟨`eos_v`⟩ is generated.

5.3 Evaluation

This section describes experiments conducted to evaluate the performance of the proposed model for AST.

5.3.1 Experimental Data

To evaluate our model, we used 54 popular songs with reliable annotations from the RWC Music Database [83]. We split the audio signals of isolated singing voice and the corresponding time-aligned musical scores [84] into all possible segments ranging from 1 measure to 8 measures with an overlap of 1 measure. When a note crossed a bar line, it was included in the precedent measure. Rests in a measure were concatenated into a single rest. Musical notes longer than a whole note were discarded. We used 44, 5, and 5 songs as training, validation, and test data, respectively.

All songs were sampled at 44.1 kHz, and we used an STFT with a Hann window of 2048 points and a shifting interval of 441 points (10 ms) for calculating magnitude spectrograms, which we normalized to make the maximum value 1. Since a tempo determines note values, which would be difficult to correctly estimate for the proposed method without any mechanism to estimate tempos, all songs were modified to a BPM of 150 using a phase vocoder for the training, validation, and test data. We standardized the spectrograms for each frequency bin, and then calculated the mel-scale spectrograms with 229 channels.

5.3.2 Configurations

The vocabulary of pitches consisted of a rest, ⟨sos_p⟩, ⟨eos_p⟩, and 40 semitone-level pitches from E2 to G5 ($K = 43$). The vocabulary of note values consisted of ⟨sos_v⟩, ⟨eos_v⟩, and 16 values which were integer multiples of a 16th note up to a whole note ($L = 18$). We discarded any data containing out-of-vocabulary notes, and it was also assumed that an input sung melody could be represented as a monophonic sequence of musical notes.

We applied frame stacking [102] with a stack size of 5 and a skip size of 1 to the mel-scale spectrograms, and added zero frames at the both ends to align with ⟨sos⟩s and ⟨eos⟩s. The encoder consisted of five-layers of bidirectional LSTMs with 300×2 cells. We set the dropout rate to 0.2 for each layer. The decoder consisted of one-layer LSTM with 200 cells. The number of channels and the filter size were $C = 10$ and $F = 100$. We used a padding size and a stride

Table 5.1: Word error rates on the test data.

Configuration of λ	NER [%]	PER [%]	VER [%]
$\lambda = 1$	48.8	31.6	34.8
$\lambda = 0$	119.8	90.0	96.8
λ is gradually reduced	43.0	24.7	29.6

of convolution in the attention mechanism of 50 and 1, respectively. Adam [103] with a standard setting was used to optimize the proposed model. The hyperparameter A was 200. To avoid overfitting, a weight decay (L2 regularization) with a controllable hyperparameter of 10^{-5} was used. To prevent weight parameters from diverging, gradient clipping with a threshold of 5.0 was also used. All weight parameters of fully-connected layers were initialized with random values drawn from the uniform distribution $\mathcal{U}(-0.1, 0.1)$. The filter of the one dimensional CNN in the attention mechanism and the weight parameters of the encoder and decoder were initialized by He's method [98]. All bias parameters were initialized with zeros. The small positive value ε was set to 10^{-4}. The batch size and the number of epochs were 20 and 15. Pytorch v$0.4.1$ [104] was used for implementation.

To verify the effectiveness of $\mathcal{L}^{\mathrm{Att}}$, we compared the following three configurations of the hyperparameter λ:

- λ is fixed to 1.
- λ is fixed to 0.
- λ is initialized to 1 and reduced by 10^{-4} at each iteration.

We also examined the effectiveness of the semi-supervised learning when only a part of $\mathbf{Z}$ is available. We randomly selected data from $\mathbf{Z}$ at a rate of 5%, 10%, 25%, 50%, and 75%, and only the selected data were used to calculate the guiding ground-truth weights $\boldsymbol{\beta}$ and the loss function $\mathcal{L}^{\mathrm{Att}}$ with a gradually reduced λ. For evaluation, we used the model which minimized the average of validation losses per an epoch during training.

Table 5.2: Note-level error rates on the test data.

Configuration of λ	E_{p} [%]	E_{m} [%]	E_{e} [%]	E_{on} [%]	E_{off} [%]	E_{all} [%]
$\lambda = 1$	14.9	8.7	9.2	19.4	16.7	13.8
$\lambda = 0$	14.7	13.4	29.3	38.9	28.0	24.8
λ is gradually reduced	13.9	6.5	10.5	18.1	16.4	13.1

5.3.3 Evaluation Metrics

Performance was measured using word error rate (WER) defined as

$$\mathrm{WER} = \frac{N_S + N_D + N_I}{N} \times 100\ [\%], \tag{5.7}$$

where the numerator represents the Levenshtein distance between ground-truth and estimated note sequences. N_S, N_D, and N_I indicate the minimum number of substitutions, deletions, and insertions required to change the estimated sequence into the ground-truth. N indicates the number of ground-truth musical notes. We used three variants of WER: note error rate (NER) for evaluating both pitches and note values, pitch error rate (PER) for evaluating only pitches, and value error rate (VER) for evaluating only note values. To measure the note-level performance, in addition we used the metrics proposed in [50] that calculate the following values: pitch error rate E_{p}, missing note error rate E_{m}, extra note rate E_{e}, onset-time error rate E_{on}, offset-time error rate E_{off}, and overall (average) error rate E_{all}.

5.3.4 Experimental Results

Experimental results are shown in Tables 5.1 and 5.2. The models obtained using the proposed loss function clearly outperformed those obtained with $\lambda = 0$. Fig. 5.2 shows NERs calculated on the validation data during the training of each proposed model. It indicates that the proposed loss function was effective in accelerating the convergence of NERs and finding a better solution with lower validation loss. An interesting fact is that better WERs and E_{all} were obtained by gradually reducing the value of λ than by fixing λ to 1. Since the time-aligned onset annotations and guiding ground-truth weights $\boldsymbol{\beta}$ were not always

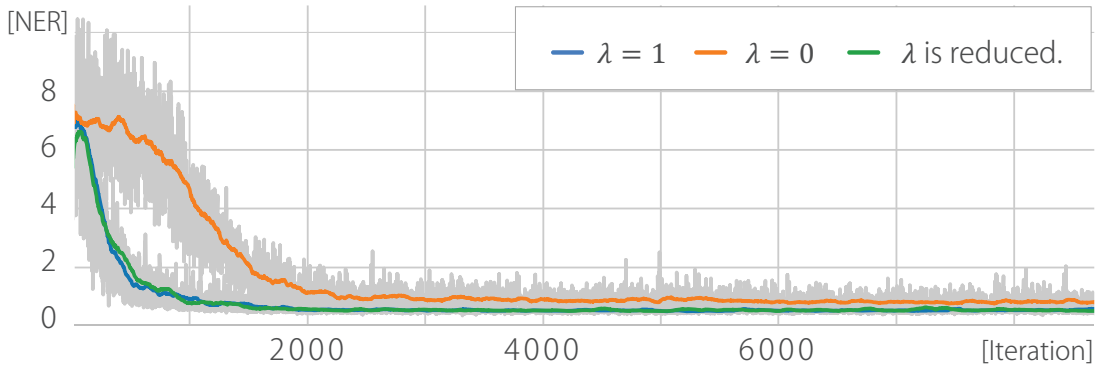


Figure 5.2: NERs calculated on the validation data during the training. Grey lines indicate NERs of each iteration, and colored lines indicate the average values of the NERs for the past 100 iterations.

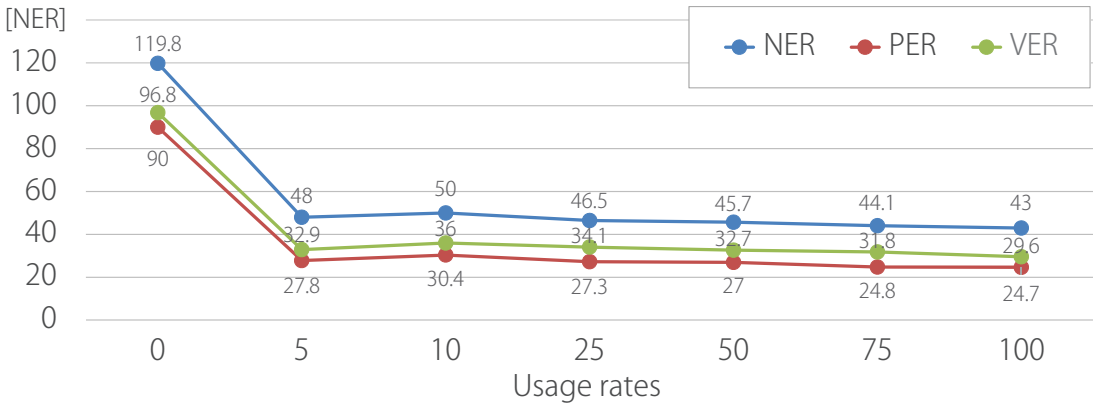


Figure 5.3: WERs with different usage rates of training data $\mathbf{Z}$.

accurate, gradually reducing λ enables the proposed model to automatically find better time alignment.

Examples of attention weights and musical notes estimated by the proposed model are illustrated in Fig. 5.4. The yellow score estimated with $\lambda = 0$ included many 8th notes that were not in the ground-truth score, and vague attention weights were learned. On the other hand, the blue and green scores estimated using the proposed loss function were nearly correct, and peaked attention weights were learned. Given that musical note estimation failed with softly-learned attention weights, it seems to be reasonable to use the peaked ground-truth weights. Although a rest that was not in the ground-truth score was estimated near the center of the bottom score, the input spectrogram certainly had an unvoiced interval near the 300th frame. The annotated data regards the

silence as a continuation of the musical note, whereas the proposed method estimated a rest for the silent section of the spectrogram. Offset detection is still an open problem in music transcription, and we will consider offsets further in future work.

Experimental results with different usage rates of $\mathbf{Z}$ are shown in Fig. 5.3. Note that the cases of using 0% and 100% of $\mathbf{Z}$ in Fig. 5.3 correspond to the second and third rows of Table 5.1, respectively. As shown in Fig. 5.3, the WERs were almost monotonically reduced as the usage rate of $\mathbf{Z}$ was increased. Especially, the WERs were drastically reduced even if only 5% of $\mathbf{Z}$ was used. This indicates that the loss function $\mathcal{L}^{\mathrm{Att}}$ is very effective even when a small amount of supervised data is available.

5.4 Summary

This chapter has presented the method for estimating the musical notes of a sung melody with an attention-based encoder-decoder model. We extended the standard model to simultaneously predict a pitch and a note value at each step. We also proposed a new loss function for attention weights and a semi-supervised training method for better performance and faster convergence. The experimental results showed that the proposed encoder-decoder model has great potential for end-to-end AST, and that the performance of AST was improved by using the attention loss function.

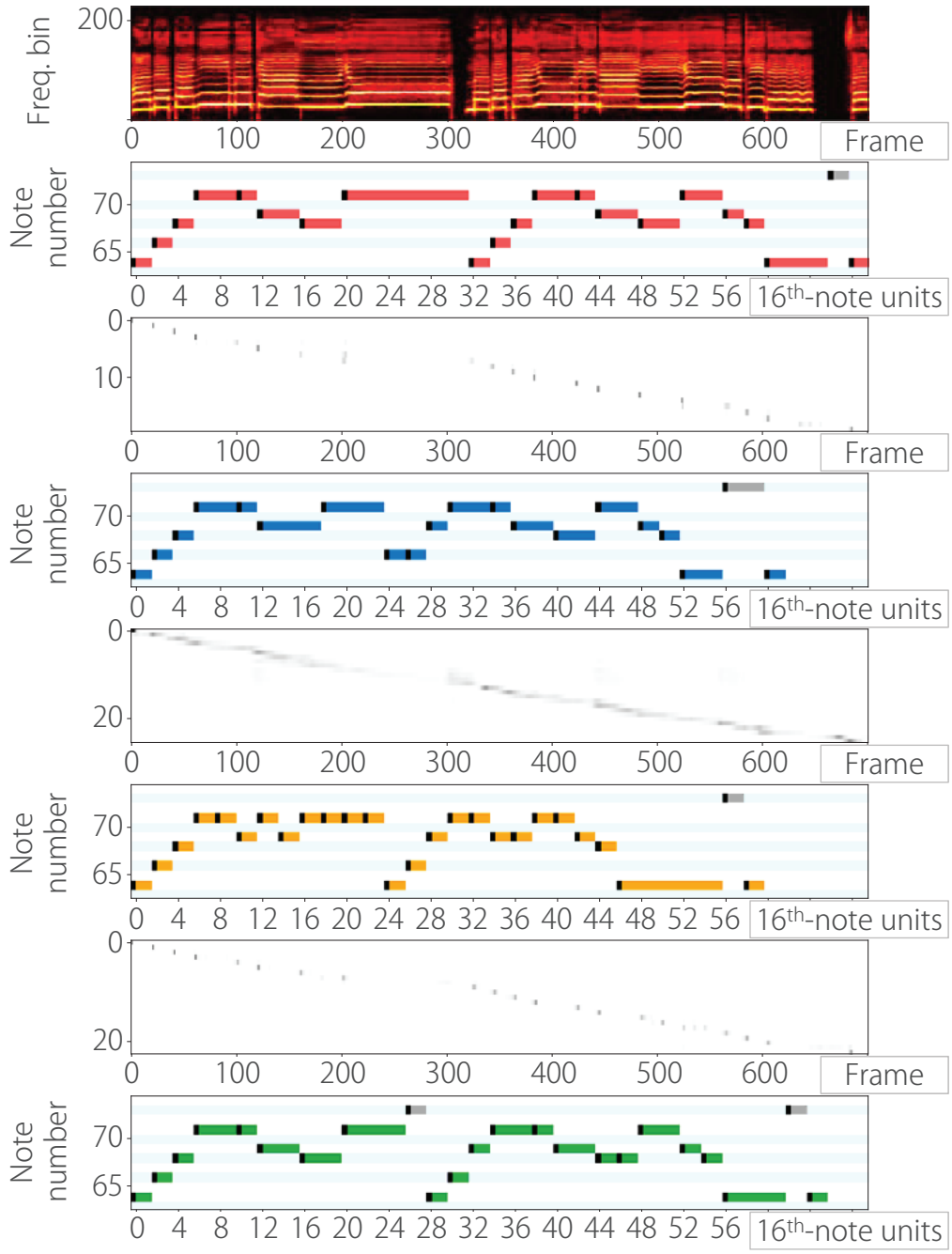


Figure 5.4: Examples of attention weights and musical notes estimated by the proposed method. Red, blue, yellow, and green horizontal lines indicate musical notes, grey lines indicate rests, and black squares indicate the onset positions of the musical notes. The top two figures are the input spectrogram and the ground-truth musical notes. The subsequent figures are attention weights and musical notes for $\lambda = 1$, $\lambda = 0$, and the gradual reduction of λ from top to bottom.

Chapter 6

Discriminative Approach Based on Encoder-Decoder Model with Tatum-Level Output

This chapter presents the discriminative approach to AST based on the encoder-decoder model. To jointly estimate musical notes and metrical structure, we propose the tatum-level decoder instead of the note-level encoder proposed in Chapter 5. In addition, we propose the beat-synchronous attention mechanism that guides an attention matrix without using time-aligned data of input signals and output symbols.

6.1 Introduction

Since AST is a typical sequence-to-sequence task that maps a sequence of acoustic features to a sequence of musical note symbols, the end-to-end approach based on a neural encoder-decoder model with an attention mechanism is considered to be promising in theory. In general, an encoder is used for converting an input sequence into a sequence of latent representations of the same length. A decoder is then used for sequentially predicting an output sequence of an appropriate length from the latent sequence while associating each output symbol with input symbols (frames) in the attention mechanism. While this approach has been investigated intensively and made a big success in machine translation [66, 67] and automatic speech recognition (AST) [69–71], only a few attempts have been

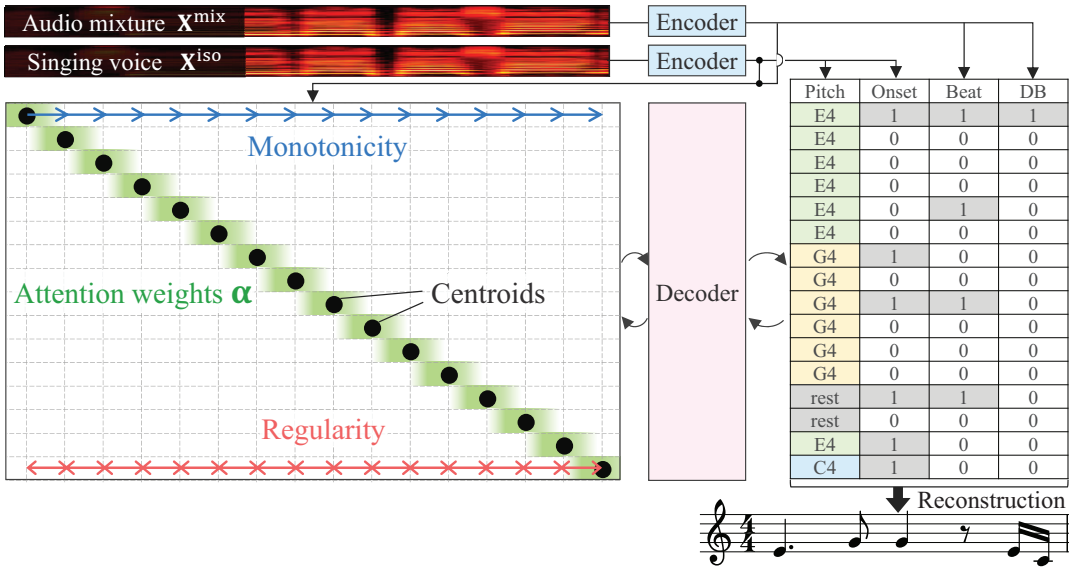


Figure 6.1: The proposed neural encoder-decoder model with a beat-synchronous attention mechanism for end-to-end singing transcription. DB = 'downbeat'. New loss functions for the centroids of attention weights are introduced to align them with equally-spanned beat times.

conducted for automatic music transcription [91].

If we limit our focus to monophonic music transcription, the fundamental difficulty of attention-based AST lies in estimation of temporal information of musical notes (metrical positions of note onsets and note values). Previously, a standard attention-based model consisting of a frame-level encoder and a *note-level* decoder was used for AST [91]. This model was found to be able to estimate note pitches, but often failed to estimate note values. This is because that the encoder is good at extracting instantaneous features (pitch and timbral information) from an input sequence, but poor at extracting temporal features (duration information). Even long short-term memory networks (LSTMs) cannot propagate temporal information through several tens or hundreds of frames that a note duration would have.

To solve this problem, we consider tatums (16th-note-level beat times) as output symbols instead of musical notes and propose a new attention-based model consisting of two frame-level encoders followed by a *tatum-level* decoder (Fig. 6.1). The two encoders independently extract latent representations about

notes and beats from singing signals (assumed to be separated in advance) and music signals, respectively. These representations are used jointly for calculating the attention weights for an output symbol at each step. The decoder sequentially predicts output symbols, *i.e.*, tatum-level score segments consisting of note pitches, note onset flags, and beat and downbeat flags. This architecture is thus considered to be able to make pitch- and beat-aware accurate input-output alignment by leveraging the metrical structures of notes and beats.

We further propose a beat-synchronous attention mechanism that imposes constraints on the attention weights in terms of *monotonicity* and *regularity*. Since popular songs typically have steady tempo and regular beats, these constraints guide together the attention centroids of output symbols to line up in ascending order with an almost equal interval. Conventional methods implement the monotonicity constraint by modifying network architectures [26] or designing a special architecture of calculating attention weights [25], whereas we implement both monotonicity and regularity constraints in the loss functions that are minimized jointly with the cross-entropy loss for output symbols.

The main contribution of this chapter is to propose a new attention model with a tatum-level decoder for popular music having regular metrical structure. Our model can jointly perform note estimation with beat and downbeat tracking in a unified framework. We experimentally investigate the effectiveness of the monotonicity and regularity constraints.

6.2 Method

The proposed method of AST is based on an encoder-decoder model with an attention mechanism (Fig. 6.1). The proposed model has two encoders that take as inputs singing and mixture spectrograms, respectively, and output latent vectors for each frame. These latent vectors are then input to a decoder with a beat-synchronous attention mechanism to output tatum-level score segments.

6.2.1 Problem Specification

The input is a music audio signal and the output is a symbolic score of the vocal part. Let T, F, and N be the number of time frames, that of frequency bins, and that of the 16th-note-level time indices, respectively. The inputs for the proposed network are the mel-scale spectrogram of an isolated singing voice $\mathbf{X}^{\mathrm{iso}} = [\mathbf{x}_1^{\mathrm{iso}}, \ldots, \mathbf{x}_T^{\mathrm{iso}}] \in \mathbb{R}_+^{F \times T}$ and that of a mixture of singing voice and accompaniment $\mathbf{X}^{\mathrm{mix}} = [\mathbf{x}_1^{\mathrm{mix}}, \ldots, \mathbf{x}_T^{\mathrm{mix}}] \in \mathbb{R}_+^{F \times T}$, where $\mathbf{x}_t^{\mathrm{iso}}$ and $\mathbf{x}_t^{\mathrm{mix}}$ indicate the mel-scale spectra at frame t of isolated and mixture signals. The isolated singing voice $\mathbf{X}^{\mathrm{iso}}$ is used as an input to simplify the problem because it is difficult to directly estimate the pitch and onset flag from the mixture signal that includes accompaniment sounds. The output of the network is a sequence of symbols $\mathbf{Y} = [y_1, \ldots, y_N]$. Each symbol $y_n = (p_n, o_n, b_n, d_n)_n$ consists of four symbols: a semitone-level pitch $p_n \in \{1, \ldots, K, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{p}}$, where K represents the size of the pitch vocabulary (including the rest), an onset flag $o_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{o}}$, a beat flag $b_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{b}}$, and a downbeat flag $d_n \in \{0, 1, \langle \mathtt{sos} \rangle, \langle \mathtt{eos} \rangle\} = V^{\mathrm{d}}$. We can reconstruct a score from the information contained in $\mathbf{Y}$. The special elements, $\langle \mathtt{sos} \rangle$ and $\langle \mathtt{eos} \rangle$, in the vocabulary of each symbol represent the start and end of an output sequence.

6.2.2 Frame-level Encoders

The proposed model has two encoders for $\mathbf{X}^{\mathrm{iso}}$ and $\mathbf{X}^{\mathrm{mix}}$. The encoders transform the spectrograms into sequences of intermediate representation vectors $\mathbf{H}^{\mathrm{iso}} = [\mathbf{h}_1^{\mathrm{iso}}, \ldots, \mathbf{h}_T^{\mathrm{iso}}] \in \mathbb{R}^{E \times T}$ and $\mathbf{H}^{\mathrm{mix}} = [\mathbf{h}_1^{\mathrm{mix}}, \ldots, \mathbf{h}_T^{\mathrm{mix}}] \in \mathbb{R}^{E \times T}$, where E is the dimension of the intermediate representation vectors. Since the length of the input spectrogram is variable, we use as each of the encoders a recurrent neural network (RNN), specifically a multi-layer bidirectional LSTM network.

6.2.3 Tatum-level Decoder with an Attention Mechanism

The decoder predicts a sequence $\mathbf{Y}$ from the latent vectors $\mathbf{H} = \mathbf{h}_{1:T} \in \mathbb{R}^{2E \times T}$, where $\mathbf{h}_t$ is a concatenation of the intermediate vectors $\mathbf{h}_t^{\mathrm{iso}}$ and $\mathbf{h}_t^{\mathrm{mix}}$. The decoder

consists of a unidirectional LSTM and is defined as follows:

$$\boldsymbol{\alpha}_n = \mathrm{Attend}(\mathbf{s}_{n-1}, \boldsymbol{\alpha}_{n-1}, \mathbf{H}), \tag{6.1}$$

$$\mathbf{g}_n = \sum_{t=1}^{T} \alpha_{nt} \mathbf{h}_t, \tag{6.2}$$

$$y_n = \mathrm{Generate}(\mathbf{s}_{n-1}, \mathbf{g}_n), \tag{6.3}$$

$$\mathbf{s}_n = \mathrm{Recurrency}(\mathbf{s}_{n-1}, \mathbf{g}_n, y_n), \tag{6.4}$$

where $\boldsymbol{\alpha}_n \in \mathbb{R}^T$ is a set of attention weights, $\mathbf{s}_n \in \mathbb{R}^D$ denotes the n-th hidden state of the decoder, and the functions Attend, Generate, and Recurrency are explained in the following.

(6.1) and (6.2) represent the attention mechanism. $\boldsymbol{\alpha}_n \in \mathbb{R}^T$ is a vector of normalized probabilities representing the degrees of relevance between the latent vectors $\mathbf{H}$ and each hidden state $\mathbf{s}_n$. Each element of $\boldsymbol{\alpha}_n$ is calculated as follows:

$$\alpha_{nt} = \frac{\exp(e_{nt})}{\sum_{t'=1}^{T} \exp(e_{nt'})}, \tag{6.5}$$

$$e_{nt} = \mathrm{Score}(\mathbf{s}_{n-1}, \mathbf{h}_t, \boldsymbol{\alpha}_{n-1}), \tag{6.6}$$

where Score is a function that calculates a raw weight. We calculate a shared matrix of attention weights $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{1:N} \in \mathbb{R}^{N \times T}$ from $\mathbf{H}^{\mathrm{iso}}$ and $\mathbf{H}^{\mathrm{mix}}$ so that the attention mechanism can put emphasis on the intermediate representations that are important in terms of both note and beat structures. We use as the Score function a convolutional function [69] given by

$$\mathbf{f}_n = \mathbf{F} * \boldsymbol{\alpha}_{n-1}, \tag{6.7}$$

$$e_{nt} = \mathbf{w}^\top \tanh\left(\mathbf{W}\mathbf{s}_{n-1} + \mathbf{V}\mathbf{h}_t + \mathbf{U}\mathbf{f}_{nt} + \mathbf{b}\right), \tag{6.8}$$

where "$*$" means the 1-dimensional convolution operation, $\mathbf{F} \in \mathbb{R}^{C \times I}$ is a set of convolution kernels, $\mathbf{f}_n \in \mathbb{R}^{C \times T}$ is the result of the convolution, and C and I indicate the number of kernels and the size of each kernel. $\mathbf{w} \in \mathbb{R}^A$ is a weight vector, $\mathbf{W} \in \mathbb{R}^{A \times D}$, $\mathbf{V} \in \mathbb{R}^{A \times 2E}$, and $\mathbf{U} \in \mathbb{R}^{A \times C}$ are weight matrices, $\mathbf{b} \in \mathbb{R}^A$ is a bias vector, and A is the number of rows of $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{U}$, as well as the number of elements of $\mathbf{b}$.

Let $\mathbf{g}_n^{\mathrm{iso}} = [g_{n1}, \ldots, g_{nE}]^\top$ and $\mathbf{g}_n^{\mathrm{mix}} = [g_{n,E+1}, \ldots, g_{n,2E}]^\top$ denote the parts of $\mathbf{g}_n$ calculated from $\mathbf{h}^{\mathrm{iso}}$ and $\mathbf{h}^{\mathrm{mix}}$ in (6.2). The generation process of y_n in (6.3) is given by

$$\boldsymbol{\phi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{p}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{p}}\mathbf{g}_n^{\mathrm{iso}} + \mathbf{b}^{\mathrm{p}}\right), \tag{6.9}$$

$$p_n = \underset{p \in V^{\mathrm{p}}}{\mathrm{argmax}} \left(\phi_p^{(n)}\right), \tag{6.10}$$

$$\boldsymbol{\psi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{o}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{o}}\mathbf{g}_n^{\mathrm{iso}} + \mathbf{b}^{\mathrm{o}}\right), \tag{6.11}$$

$$o_n = \underset{o \in V^{\mathrm{o}}}{\mathrm{argmax}} \left(\psi_o^{(n)}\right), \tag{6.12}$$

$$\boldsymbol{\eta}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{b}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{b}}\mathbf{g}_n^{\mathrm{mix}} + \mathbf{b}^{\mathrm{b}}\right), \tag{6.13}$$

$$b_n = \underset{b \in V^{\mathrm{b}}}{\mathrm{argmax}} \left(\eta_b^{(n)}\right), \tag{6.14}$$

$$\boldsymbol{\xi}^{(n)} = \mathrm{softmax}\left(\mathbf{P}^{\mathrm{d}}\mathbf{s}_{n-1} + \mathbf{Q}^{\mathrm{d}}\mathbf{g}_n^{\mathrm{mix}} + \mathbf{b}^{\mathrm{d}}\right), \tag{6.15}$$

$$d_n = \underset{d \in V^{\mathrm{d}}}{\mathrm{argmax}} \left(\xi_d^{(n)}\right), \tag{6.16}$$

where $\mathbf{P}^\star \in \mathbb{R}^{|V^\star| \times D}$, $\mathbf{Q}^\star \in \mathbb{R}^{|V^\star| \times E}$ are weight matrices, and $b^\star \in \mathbb{R}^{|V^\star|}$ is a bias parameter. Here, "$\star$" represents "p" (pitch), "o" (onset), "b" (beat), or "d" (downbeat). Note that $\mathbf{g}_n^{\mathrm{iso}}$ is used for estimating the pitches and onsets that are components of musical notes of a sung melody and $\mathbf{g}_n^{\mathrm{mix}}$ is used for estimating the beats and downbeats from the percussive sounds in the mixture sound.

(6.4) represents the recursive calculation of the next state $\mathbf{s}_n$. We adopt the teacher forcing for training. In short, the concatenation of one-hot vectors separately converted from the ground-truth pitch, onset, beat, and downbeat is used as y_n. The proposed model is optimized by minimizing the sum of the cross entropies for the elements of each y_n and the additional losses described in the next subsection. In the inference phase, the output symbols obtained by (6.10), (6.12), (6.14), and (6.16) at the previous step are converted into one-hot vectors and used for the concatenation of the one-hot vectors for predicting the current symbol. This recursive process is stopped when the output sequence reaches a predefined maximum length or when $\langle \mathsf{eos} \rangle$ symbol is generated as p_n, o_n, b_n, or d_n.

6.2.4 Loss Functions for Attention Weights

We introduce new loss functions for the attention weights $\boldsymbol{\alpha}_{1:N} \in \mathbb{R}^{N \times T}$ to satisfy the *monotonicity* and *regularity* constraints mentioned in Section 6.1. In appropriate input-output alignment, the attention weights of each $\boldsymbol{\alpha}_n$ are known to be biased toward a narrow region in the time axis. As a representative point of each $\boldsymbol{\alpha}_n$, we use the centroid given by

$$c_n = \sum_{t=1}^{T} t \cdot \alpha_{nt}. \tag{6.17}$$

The loss function regarding monotonicity is given by

$$\mathcal{L}^{\text{mono}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \text{ReLU}\left(-\Delta c_n\right), \tag{6.18}$$

where $\Delta c_n = c_{n+1} - c_n$ is the difference of the consecutive centroids, and ReLU is a rectified linear function given by $\text{ReLU}\left(x\right) = \max\left(0, x\right)$. (6.18) prevents the order of the centroids from being reversed by imposing the positive cost only if the order of adjacent centroids is reversed.

The loss function regarding regularity is given by

$$\mathcal{L}^{\text{reg}} = \frac{1}{N-2} \sum_{n=1}^{N-2} \left|\Delta c_{n+1} - \Delta c_n\right|^2. \tag{6.19}$$

This function makes the centroids be arranged at almost equal intervals that do not suddenly change over time.

6.3 Evaluation

This section reports the results of comparative evaluations on the performance of the proposed method.

6.3.1 Data

To evaluate our model, we used 54 popular songs with reliable annotations from the RWC Music Database [83]. We split the input audio signals and the

Table 6.1: Error rates [%] in tatum and note levels.

Method	Attention loss		Tatum-level error rate				
	Monotonicity	Regularity	$E_{\mathrm{p}}^{\mathrm{T}}$	$E_{\mathrm{o}}^{\mathrm{T}}$	$E_{\mathrm{b}}^{\mathrm{T}}$	$E_{\mathrm{d}}^{\mathrm{T}}$	$E_{\mathrm{a}}^{\mathrm{T}}$
Proposed	✓	✓	67.5	29.4	15.7	18.1	77.2
	✓		42.6	26.0	16.4	18.8	58.2
			34.6	22.5	15.4	17.8	48.3
Majority-vote	n/a	n/a	34.8	25.1	n/a	n/a	n/a
Method	Attention loss		Note-level error rate [50]				
	Monotonicity	Regularity	$E_{\mathrm{p}}^{\mathrm{N}}$	$E_{\mathrm{m}}^{\mathrm{N}}$	$E_{\mathrm{e}}^{\mathrm{N}}$	$E_{\mathrm{on}}^{\mathrm{N}}$	$E_{\mathrm{off}}^{\mathrm{N}}$
Proposed	✓	✓	19.5	43.7	67.5	61.1	48.8
	✓		20.1	20.9	42.2	55.8	42.0
			18.3	11.4	31.7	46.3	34.1
Majority-vote	n/a	n/a	20.4	11.3	51.3	55.5	51.7

corresponding tatum-level scores into segments of 8 secs with an overlap of 1 sec. When we generated the tatum-level scores and split them, we referred to the annotated musical scores and beat times [84]. If the tatum crossed the start boundary of the segment, the tatum was removed from the segment. The tatum crossing the end boundary of the segment remained in the segment. We did not use segments including only rests.

All songs were sampled at 44.1 kHz, and we used a short-time Fourier transform (STFT) with a Hann window of 2048 points and a shifting interval of 441 points (10 ms) for calculating magnitude spectrograms, which were normalized to make the maximum value equal to 1, followed by the calculation of the mel-scale spectrograms with 229 channels. The mel-scale spectrograms of training data were standardized for each frequency bin, and the mel-scale specgtrograms of evaluation data were standardized for each frequency bin by using means and standard deviations calculated from the training data.

6.3.2 Setup

The vocabulary of pitches consisted of rest, 40 semitone-level pitches from E2 to G5 ($K = 41$), and the special elements $\langle \mathtt{sos} \rangle$ and $\langle \mathtt{eos} \rangle$. The pitch vocabulary

covered pitches contained in both the train and test data. We assumed that an input sung melody can be represented as a monophonic sequence of musical notes.

Each encoder consisted of a three-layer bidirectional LSTM with 100×2 cells with the dropout rate of 0.2. The decoder consisted of one-layer LSTM with 100 cells. A padding size and a stride of convolution in the attention mechanism were set to 50 and 1, respectively, and the parameters were $C = 10$, $I = A = 100$. Adam [103] was used to optimize the parameters of the proposed model, and a weight decay (L2 regularization) with a controllable hyperparameter 10^{-5} and a gradient clipping with a threshold of 5.0 were applied for training. All weight parameters of fully-connected layers were initialized with random values drawn from the uniform distribution $\mathcal{U}(-0.1, -0.1)$, and all bias parameters were initialized with zeros. The kernels of the CNN in the attention mechanism and the weight parameters of the encoder and decoder were initialized by He's method [98]. The maximum lengths in the inference phase was set to 200. The batch size and the number of epochs were 150 and 100, respectively. PyTorch v1.0.1 was used for implementation.

6.3.3 Metrics

The performance of tatum-level transcription was measured using tatum-level error rate defined as:

$$\frac{N_S + N_D + N_I}{N} \times 100\ [\%], \tag{6.20}$$

where the numerator represents the Levenshtein distance between the ground-truth and prediced scores: N_S, N_D, and N_I are the minimum number of substitutions, deletions, and insertions required to change the predicted sequence into the ground-truth N is the number of tatums in the ground-truth. We used the error rate for evaluating each of a pitch $E^{\mathrm{T}}_{\mathrm{p}}$, an onset flag $E^{\mathrm{T}}_{\mathrm{o}}$, a beat flag $E^{\mathrm{T}}_{\mathrm{b}}$, and a downbeat flag $E^{\mathrm{T}}_{\mathrm{d}}$ and all of them $E^{\mathrm{T}}_{\mathrm{a}}$.

To measure the note-level performance, we used the metrics proposed in [50] that calculate the following five values: pitch error rate $E^{\mathrm{N}}_{\mathrm{p}}$, missing note error

rate $E_{\mathrm{m}}^{\mathrm{N}}$, extra note rate $E_{\mathrm{e}}^{\mathrm{N}}$, onset-time error rate $E_{\mathrm{on}}^{\mathrm{N}}$, and offset-time error rate $E_{\mathrm{off}}^{\mathrm{N}}$. In constructing of the score from the predicted symbols, we applied the following rules:

1. If $p_{n-1} \neq p_n$, then the $(n-1)$-th and n-th tatums are included in different notes.
2. If $p_{n-1} = p_n$ and $o_n = 1$, then the $(n-1)$-th and n-th tatums are included in different notes having the same pitch.
3. If $p_{n-1} = p_n$ and $o_n = 0$, then the $(n-1)$-th and n-th tatums are included in the same note.

Note that the note-level metrics do not take into account rests, beats and downbeats. We used the parameters that minimize an epoch average of E_p^T calculated using some of the evaluation data.

6.3.4 Results

Experimental results in Table 6.1 showed that both tatum- and note-level error rates became worse by using the proposed loss functions for attention weights. The reason for this result is that the constraint of the loss functions is too strong to prevent the attention mechanism from finding appropriate position in the input sequence. In the early stages of training, the centroids usually line up at the roughly same positions, and $\mathcal{L}^{\mathrm{mono}}$ and $\mathcal{L}^{\mathrm{reg}}$ are almost zero. The centroids cannot escape from the initial positions because the values of $\mathcal{L}^{\mathrm{mono}}$ and $\mathcal{L}^{\mathrm{reg}}$ increase when the centroids change.

We also compared the proposed method to the majority-vote method used in [78] using ground-truth F0 trajectory with voice activity detection and tatum times. A boundary of pitch changes was regarded as note onset positions and the successive tatums having the same pitch were included in one note. Since the method does not estimate beats and downbeats, $E_{\mathrm{b}}^{\mathrm{T}}$, $E_{\mathrm{d}}^{\mathrm{T}}$, and $E_{\mathrm{a}}^{\mathrm{T}}$ are not used. In most metrics, the proposed method without the attention losses attained better error rates than the majority-vote method. Especially, we obtained an improvement on E_e^N.

6.4 Summary

This chapter presented the method for musical note transcription of a sung melody based on the encoder-decoder model with the beat-synchronous attention mechanism. We extended the standard encoder-decoder model to simultaneously predict a pitch, an onset flag, a beat flag, and a downbeat flag that are needed to construct a musical score. We also proposed the loss functions to induce the attention weights to have proper structure. We experimentally investigated whether those loss functions are effective or not.

Chapter 7

Conclusion

This thesis addressed audio-to-score singing transcription for popular music. We summarize the contributions of this thesis and discuss the future directions.

7.1 Contributions

In Chapter 3, we presented a *generative* approach estimates note sequences from vocal F0 trajectories with tatum times. The proposed approach was based on a hierarchical hidden semi-Markov model (HSMM) that integrates a *generative* language model describing the transitions of local keys and the rhythms of notes with a *generative* semi-acoustic model precisely describing the temporal-frequency deviations of the F0 trajectory. Given an F0 trajectory and beat times, a sequence of musical notes, that of local keys, and the temporal and frequency deviations can be estimated jointly by using a Markov chain Monte Carlo (MCMC) method. We experimentally confirmed that the performances of AST improved by considering both local keys and onset score times in the language model. On the other hand, the acoustic model often degraded the AST performances because the large deviations in the F0 trajectory were hard to represent precisely and the acoustic model sometimes overfit singing expressions like overshoot and preparation.

In Chapter 4, we presented a *hybrid* approach to the estimation of note sequences from vocal spectrograms with tatum times. The proposed approach is based on an HSMM of a vocal spectrogram that consists of a *generative* language

model similar to that in Chapter 3 and a *discriminative* acoustic model based on a convolutional recurrent neural network (CRNN) trained in a supervised manner for predicting the posterior probabilities of note pitches and onsets from the spectrogram. Thanks to the CRNN extracting effective singing features from music signals, musical notes including rests and consecutive notes of the same pitch can be directly and accurately estimated without using F0 estimation. Given a vocal spectrogram and beat times, the most-likely note sequence is estimated with the Viterbi algorithm while leveraging both the grammatical knowledge about musical notes and the expressive power of the CRNN. We experimentally showed that the proposed CRNN-HSMM achieved the state-of-the-art performance thanks to the effective combination of the key- and rhythm-aware regularization of the estimated note sequence and the robustness of the CRNN against the large variation of singing voice. We also confirmed that further refinement of the voice separation method and beat-tracking method could improve the transcription results by the hybrid approach.

In Chapters 5, we presented *discriminative* approaches based on the encoder-decoder architecture with the attention mechanism, where the encoder and decoder are considered to work as *discriminative* acoustic and language models, respectively. The proposed model consisted of a frame-level encoder and a *note-level* decoder for directly estimating a variable-length sequence of notes (pitches and note values) from a vocal spectrogram. To make effective use of fewer aligned data for stabilizing the training process, we proposed an attention loss that leads the frame-wise attention weights of each note to concentrate on the ground-truth onset frame. In addition, by gradually reducing the weight of the attention loss, better input-output alignment can be learned faster. We experimentally showed the effectiveness of the attention loss with gradual reduction. We also confirmed that the performances drastically improved even if only a small amount of the time-aligned data.

In Chapter 6, we proposed a new encoder-decoder model consisting of a frame-level encoder and a *tatum-level* decoder for directly estimating sequences of pitches and binary onset activations, both of which are instantaneous at-

tributes, at the tatum level. By representing the decoder outputs at the tatum level instead of the note level, the model can jointly predict metrical structure (*i.e.*, beat and downbeat activations) in addition to the pitches and the onset activations. In addition, we proposed a beat-synchronous attention mechanism for constraining the attention weight to fulfill the property called monotonicity and regularity. We experimentally reported the performance and remaining problems of the proposed method.

7.2 Future Work

This section describes the remaining open problems regarding audio-to-score AST. All the proposed AST methods described in this thesis have been developed under the assumption that the tatum interval (temporal resolution of musical notes on musical scores) is a 16th-note length and thus cannot deal with irregular notes (*e.g.*, triplets) and shorter notes whose durations are not given by integer multiples of the 16th-note length. A naive solution to this problem is to set the tatum interval to a finer unit, *e.g.*, one-third of a 32nd-note length. Because the actual onset times of notes are temporally deviated from the regular tatum-level grid by a larger number of tatum intervals, the note values tend to be over- or under-estimated.

The proposed methods have limitations regarding the treatment of metrical structure (*i.e.*, meters and downbeats). The generative and hybrid approaches described in Chapters 3 and 4 assume that the meter of a target song is 4/4 time and the downbeats are given. Popular music often includes meter changes, which have not been considered in this thesis. A possible solution is to introduce latent variables representing meters into the language model. The discriminative approach described in Chapter 5 does not estimate the meters and downbeats. The discriminative approach described in Chapter 6 often yields the beat and downbeat activations that are inconsistent with each other. It is thus necessary to develop as a network output a new representation that does not suffer from the metrical inconsistency.

The discriminative approach in Chapter 5 uses a small amount of time-aligned data of music signals and musical notes for improving the AST performances. The weakly-supervised learning framework with rough alignment can effectively guide the attention weight matrix. One possibility of automatically increasing alignment data is to utilize another RNN trained with the connectionist temporal classification (CTC). Furthermore, by integrating the CTC-based RNN with the attention-based model, we could simultaneously predict the alignment and musical notes.

Another promising direction is to integrate the attention-based encoder-decoder models described in Chapters 5 and 6 with the generative language model used in Chapters 3 and 4. The generative language model, for example, could be used as an external module of the decoder RNN as proposed in the fusion-based methods for ASR [105–107]. In addition, the explicit hierarchical structure of keys, notes, and rhythms of the generative language model might be used to impose the structural constraints on the attention weight matrix. It is also interesting to handle music with multiple vocal parts.

Bibliography

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic nusic transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] D. J. Hermes, "Measurement of pitch by subharmonic summation," *The Journal of the Acoustical Society of America*, vol. 83, no. 1, pp. 257–264, 1988.

[4] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[5] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA Journal)*, vol. 43, no. 4, pp. 311–329, 2004.

[6] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 20, no. 6, pp. 1759–1770, 2012.

[7] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 659–663, 2014.

[8] Y. Ikemiya, K. Yoshii, and K. Itoyama, "Singing voice analysis and editing based on mutually dependent F0 estimation and source separation," in

International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 574–578, 2015.

[9] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 63–70, 2017.

[10] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 161–165, 2018.

[11] T. Nakano, K. Yoshii, Y. Wu, R. Nishikimi, K. W. Edward Lin, and M. Goto, "Joint singing pitch estimation and voice separation based on a neural harmonic structure renderer," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 160–164, 2019.

[12] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 1118–1128, 2020.

[13] Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 15, no. 4, pp. 1475–1487, 2007.

[14] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 57–60, 2012.

[15] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 23–27, 2017.

[16] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 334–340, 2018.

[17] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - a

reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[18] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[19] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, p. 1313, 2017.

[20] M. Hamanaka, K. Hirata, and S. Tojo, "deepGTTM-III: Simultaneous learning of grouping and metrical structures," in *International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pp. 161–172, 2017.

[21] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 311–316, 2011.

[22] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[23] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. Springer, 2006.

[24] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python audio and music signal processing library," in *ACM International Conference on Multimedia (ACMMM)*, pp. 1174–1178, 2016.

[25] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *International Conference on Machine Learning (ICML)*, pp. 2837–2846, 2017.

[26] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *International Conference on Learning Representations (ICLR)*, pp. 1–16, 2018.

[27] A. Tjandra, S. Sakti, and S. Nakamura, "Local monotonic attention mechanism for end-to-end speech and language processing," *International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 431–440, 2017.

[28] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-

speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4788, IEEE, 2018.

[29] E. Molina, L. J. Tardón, A. M. Barbancho, and I. Barbancho, "SiPTH: Singing transcription based on hysteresis defined on the pitch-time curve," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 23, no. 2, pp. 252–263, 2015.

[30] N. Kroher and E. Gómez, "Automatic transcription of flamenco singing from polyphonic music recordings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 24, no. 5, pp. 901–913, 2016.

[31] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–86, 2008.

[32] 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 *TENOR*, pp. 23–30, 2015.

[33] L. Yang, A. Maezawa, J. B. L. Smith, and E. Chew, "Probabilistic transcription of sung melody using a pitch dynamic model," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 301–305, 2017.

[34] Z.-S. Fu and L. Su, "Hierarchical Classification Nnetworks for Singing Voice Segmentation and Transcription," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 900–907, 2019.

[35] K. O'Hanlon and M. D. Plumbley, "Polyphonic piano transcription using non-negative matrix factorisation with group sparsity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3112–3116, 2014.

[36] E. Benetos and T. Weyde, "An efficient temporally-constrained probabilistic model for multiple-instrument music transcription," pp. 701–707, 2015.

[37] T. Cheng, M. Mauch, E. Benetos, and S. Dixon, "An attack/decay model for piano transcription," in *International Society for Music Information Retrieval*

Conference (ISMIR), pp. 584–590, 2016.

[38] R. Schramm, A. Mcleod, M. Steedman, and E. Benetos, "Multi-Pitch Detection and Voice Assignment for a Cappella Recordings of Multiple Singers," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 552–559, 2017.

[39] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *International Society for Music Information Retrieval Conference*, pp. 475–481, 2016.

[40] Y.-t. Wu, B. Chen, and L. Su, "Polyphonic music transcription with semantic segmentation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 166–170, 2019.

[41] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *International Society for Music Information Retrieval Conference*, pp. 50–57, 2018.

[42] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 670–677, 2019.

[43] D. Temperley and D. Sleator, "Modeling meter and harmony: A preference-rule approach," *Computer Music Journal*, vol. 23, no. 1, pp. 10–27, 1999.

[44] P. Desain and H. Honing, "The quantization of musical time: A connectionist approach," *Computer Music Journal*, vol. 13, no. 3, pp. 56–66, 1989.

[45] M. Tanji, D. Ando, and H. Iba, "Improving metrical grammar with grammar expansion," in *Australasian Joint Conference on Artificial Intelligence*, pp. 180–191, Springer, 2008.

[46] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira, and S. Sagayama, "Hidden Markov model for automatic transcription of MIDI signals," in *IEEE Workshop on Multimedia Signal Processing*, pp. 428–431, 2002.

[47] C. Raphael, "A hybrid graphical model for rhythmic parsing," *Artificial*

Intelligence, vol. 137, no. 1-2, pp. 217–238, 2002.

[48] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based quantization: Unsupervised estimation of the model parameters," in *International Conference on Multimodal Interfaces (ICMI)*, pp. 369–372, 2003.

[49] A. T. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *Journal of Artificial Intelligence Research*, vol. 18, no. 1, pp. 45–81, 2003.

[50] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 25, no. 4, pp. 794–806, 2017.

[51] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, "Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 101–105, 2018.

[52] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," in *arXiv:2008.12710v1*, pp. 1–15, 2020.

[53] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 151–155, 2017.

[54] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Neural Information Processing Systems (NIPS)*, pp. 3104–3112, 2014.

[55] H.-W. Nienhuys and J. Nieuwenhuizen, "Lilypond, a system for automated music engraving," in *Colloquium on Musical Informatics (CIM)*, pp. 167–171, 2003.

[56] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "An end-to-end framework for audio-to-score music transcription on monophonic excerpts," in *International Society for Music Information Retrieval Conference (WASPAA)*,

pp. 34–41, 2018.

[57] M. A. Román, A. Pertusa, and J. Calvo-zaragoza, "A holistic approach to polyphonic music ttranscription with neural networks," in *International Society for Music Information Retrieval Conference (ISMIR 2019)*, pp. 731–737, 2019.

[58] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "Data representations for audio-to-score monophonic music transcription," *Expert Systems with Applications*, vol. 162, p. 113769, 2020.

[59] A.-R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, vol. 1, p. 39, 2009.

[60] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4688–4691, 2011.

[61] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 20, no. 1, pp. 14–22, 2011.

[62] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *INTERSPEECH*, pp. 2578–2581, 2012.

[63] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*, pp. 8614–8618, 2013.

[64] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 15, no. 3, pp. 982–994, 2007.

[65] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734,

2014.

[66] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421, 2015.

[67] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, pp. 1–15, 2015.

[68] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[69] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Neural Information Processing Systems (NIPS)*, pp. 577–585, 2015.

[70] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

[71] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of "attention" in sequence-to-sequence models," in *INTERSPEECH*, pp. 3702–3706, 2017.

[72] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.

[73] Y. Ojima, E. Nakamura, K. Itoyama, and K. Yoshii, "A hierarchical bayesian model of chords, pitches, and spectrograms for multipitch analysis," in *International Society for Music Information Retrieval Conference, (ISMIR)*, pp. 309–315, 2016.

[74] A. McLeod, R. Schramm, M. Steedman, and E. Benetos, "Automatic transcription of polyphonic vocal music," *Applied Sciences*, vol. 7, no. 12, 2017.

[75] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d. Garcez,

and S. Dixon, "A hybrid recurrent neural network for music transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2061–2065, 2015.

[76] A. Ycart and E. Benetos, "A study on LSTM networks for polyphonic music sequence modelling," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 421–427, 2017.

[77] E. Nakamura, R. Nishikimi, S. Dixon, and K. Yoshii, "Probabilistic sequential patterns for singing transcription," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1905–1912, 2018.

[78] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Scale- and rhythm-aware musical note estimation for vocal F0 trajectories based on a semi-tatum-synchronous hierarchical hidden semi-markov model," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 376–382, 2017.

[79] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[80] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor Processes," in *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, pp. 985–992, 2006.

[81] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

[82] A. Lee, T. Kawahara, and K. Shikano, "Julius - an open source real-time large vocabulary recognition engine," in *European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1691–1694, 2001.

[83] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical and jazz music databases," in *International Conference on Music Information Retrieval (ISMIR)*, pp. 287–288, 2002.

[84] M. Goto, "AIST annotation for the RWC Music Database.," in *International*

Conference on Music Information Retrieval (ISMIR), pp. 359–360, 2006.

[85] T. De Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 01, pp. 47–70, 2011.

[86] E. Molina, A. M. Barbancho, L. J. Tardón, and I. Barbancho, "Evaluation framework for automatic singing transcription," in *International Society for Music Information Retrieval (ISMIR)*, pp. 567–572, 2014.

[87] R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii, "Musical note estimation for F0 trajectories of singing voices based on a bayesian semi-beat-synchronous HMM," in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 461–467, 2016.

[88] E. Kapanci and A. Pfeffer, "Signal-to-score music transcription using graphical models," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 758–765, 2005.

[89] C. Raphael, "A graphical model for recognizing sung melodies," in *International Conference on Music Information Retrieval (ISMIR)*, pp. 658–663, 2005.

[90] R. Nishikimi, E. Nakamura, M. Goto, K. Itoyama, and K. Yoshii, "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and f0 trajectories," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 28, pp. 1678–1691, 2020.

[91] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 161–165, 2019.

[92] R. Nishikimi, E. Nakamura, M. Goto, and K. Yoshii, "End-to-end melody note transcription based on a beat-synchronous attention mechanism," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 26–30, 2019.

[93] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE*

Transactions on Audio, Speech, and Language Processing (TASLP), vol. 20, no. 1, pp. 30–42, 2012.

[94] "CeVIO." `http://cevio.jp`.

[95] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnAudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks," *IEEE Access*, vol. 8, pp. 161981–162003, 2020.

[96] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," in *arXiv preprint arXiv:1607.08022*, pp. 1–6, 2017.

[97] D. Misra, "Mish: A self regularized non-monotonic neural activation function," in *arXiv preprint arXiv:1908.08681*, pp. 1–14, 2019.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

[99] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[100] E. Nakamura, K. Itoyama, and K. Yoshii, "Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note patterns," in *European Signal Processing Conference (EUSIPCO)*, pp. 1946–1950, 2016.

[101] J.-L. Durrieu, G. Richard, B. David, and C. Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 18, no. 3, pp. 564–575, 2010.

[102] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *INTERSPEECH*, pp. 1468–1472, 2015.

[103] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, pp. 1–15, 2014.

[104] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[105] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *INTERSPEECH*, pp. 523–527, 2017.

[106] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv:1503.03535*, 2015.

[107] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," in *INTERSPEECH*, pp. 387–391, 2017.

List of Publications

Journal Articles

1) Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii, "Audio-to-Score Singing Transcription Based on a CRNN-HSMM Hybrid Model," *APSIPA Transactions on Signal and Information Processing*, vol. xx, pp.xxx-xxx, 2021 → **Chapter 4** (minor revision).

2) Ryo Nishikimi, Eita Nakamura, Masataka Goto, Katsutoshi Itoyama, and Kazuyoshi Yoshii, "Bayesian Singing Transcription Based on a Hierarchical Generative Model of Keys, Musical Notes, and F0 Trajectories," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp.1678-1691, 2020 → **Chapter 3**.

International Conferences

3) Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii, "End-to-End Melody Note Transcription Based on a Beat-Synchronous Attention Mechanism," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 26–30, 2019 → **Chapter 5**.

4) Ryo Nishikimi, Eita Nakamura, Satoru Fukayama, Masataka Goto, and Kazuyoshi Yoshii, "Automatic Singing Transcription Based on Encoder-Decoder Recurrent Neural Networks with a Weakly-Supervised Attention Mechanism," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 161–165, 2019 → **Chapter 6**.

5) Ryo Nishikimi, Eita Nakamura, Masataka Goto, Katsutoshi Itoyama, and

Kazuyoshi Yoshii, “Scale- and Rhythm-Aware Musical Note Estimation for Vocal F0 Trajectories Based on a Semi-Tatum-Synchronous Hierarchical Hidden Semi-Markov Model,” in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 376–382, 2017.

6) Ryo Nishikimi, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii, “Musical Note Estimation for F0 Trajectories of Singing Voices Based on a Bayesian Semi-beat-synchronous HMM,” in *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 461–467, 2016.

Domestic Conferences

7) 錦見 亮, 中村 栄太, 吉井 和佳, “ビート同期注意機構に基づく歌声のリズム採譜,” 情報処理学会 第 124 回音楽情報科学研究会, Vol. 2019-MUS-124, No. 8, pp. 1–6, 2019.

8) 錦見 亮, 中村 栄太, 深山覚, 後藤真孝 , 吉井 和佳, “注意機構を用いたエンコーダ・デコーダモデルに基づく歌声の音符推定,” 情報処理学会 第 120 回音楽情報科学研究会, Vol. 2018-MUS-120, No. 7, pp. 1–6, 2018.

9) 錦見 亮, 中村 栄太, 後藤真孝, 糸山 克寿, 吉井 和佳, “調とリズムを考慮した階層隠れセミマルコフモデルに基づく歌声の自動採譜,” 電子情報通信学会 第 20 回情報論的学習理論ワークショップ (IBIS2017), Vol. 117, No. 293, pp. 147–153, 2017.

10) 錦見 亮, 中村 栄太, 糸山 克寿, 吉井 和佳, “スケールと音高の過渡的変化を考慮した HSMM に基づく歌声 F0 軌跡に対する音符推定,” 情報処理学会 第 79 回全国大会, 2017.

11) 錦見 亮, 中村 栄太, 糸山 克寿, 吉井 和佳, “歌声 F0 軌跡に対する自動採譜のための準ビード同期セグメンタル HMM,” 電子情報通信学会 第 19 回情報論的学習理論ワークショップ (IBIS2016), Vol. 116, No. 300, pp. 337–343, 2016.

12) 錦見 亮, 中村 栄太, 糸山 克寿, 吉井 和佳, “歌声 F0 軌跡に対する音符推定のためのベイジアン準ビート同期 HMM”, 情報処理学会 第 112 回音楽情報科学研

究会, Vol. 2016-MUS-112, No. 7, pp. 1–7, 2016.

13) 錦見 亮, 中村 栄太, 糸山 克寿, 吉井 和佳, “ビート準同期隠れマルコフモデルに基づく歌声音高軌跡に対する音符推定,” 情報処理学会 第 78 回全国大会, 2016.