APSIPA Transactions on Signal and Information Processing, 2022, 11, e19

This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by-nc/4.0/), which permits unrestricted re-use, distribution, and reproduction in any medium, for non-commercial use, provided the original work is properly cited.

Original Paper

Joint Chord and Key Estimation Based on a Hierarchical Variational Autoencoder with Multi-task Learning

Yiming Wu[1] and Kazuyoshi Yoshii[1,2]*

[1] *Graduate School of Informatics, Kyoto University, Kyoto, Japan*
[2] *PRESTO, Japan Science and Technology Agency, Tokyo, Japan*

ABSTRACT

This paper describes a deep generative approach to joint chord and key estimation for music signals. The limited amount of music signals with complete annotations has been the major bottleneck in supervised multi-task learning of a classification model. To overcome this limitation, we integrate the supervised multi-task learning approach with the unsupervised autoencoding approach in a mutually complementary manner. Considering the typical process of music composition, we formulate a hierarchical latent variable model that sequentially generates keys, chords, and chroma vectors. The keys and chords are assumed to follow a language model that represents their relationships and dynamics. In the framework of amortized variational inference (AVI), we introduce a classification model that jointly infers discrete chord and key labels and a recognition model that infers continuous latent features. These models are combined to form a variational autoencoder (VAE) and are trained jointly in a (semi-)supervised manner, where the generative and language models act as regularizers for the classification model. We comprehensively investigate three different architectures for the chord and key classification model, and three different architectures for the

*Corresponding author: Kazuyoshi Yoshii, yoshii@i.kyoto-u.ac.jp. This work was partially supported by JSPS KAKENHI Nos. 16H01744, 19H04137, 19K20340, and 20K21813, JST ACCEL No. JPMJAC1602, and JST PRESTO No. JPMJPR20CB.

Received 25 October 2021; Revised 19 March 2022
ISSN 2048-7703; DOI 10.1561/116.00000052
© 2022 Y. Wu and K. Yoshii

language model. Experimental results demonstrate that the VAE-based multi-task learning improves chord estimation as well as key estimation.

Keywords: Automatic chord estimation, automatic key estimation, variational autoencoder, multi-task learning

1 Introduction

Computational music signal analysis has been one of the most fundamental research topics in the field of music information retrieval (MIR). It aims to infer musical symbols behind music signals, i.e., reproduce the human ability to understand music as a set of discrete concepts. Although human experts are capable of music transcription, automating this process has been very challenging between the acoustic and symbolic domains.

In music analysis based on deep learning, joint estimation of multiple kinds of musical elements has not received much attention so far. According to western musical theories, different musical elements that describe a piece of music are semantically related. For example, chord labels are strongly affected by the underlying key labels [22], and the chord and key transitions tend to occur at (down)beat positions [28]. Most conventional methods, however, take the discriminative approach based on supervised learning of an audio-to-label transcription process [19, 20, 26], where the mutual dependency of multiple musical elements and a label-to-audio generative process are not taken into account. Statistical representation of the complicated relationships between mutually-dependent musical elements through multi-task learning is thus considered the key to further improvement [30].

The major bottleneck of DNN-based multi-task learning lies in the limited amount of completely annotated music signals used for supervised training of a multi-label classifier [30]. It is extremely time-consuming to make time-synchronized multi-label annotations on music signals. This makes it hard to draw the full potential of highly expressive deep neural networks (DNNs). A multi-task classification model often underperforms independently- but fully-trained single-task models. This calls for a principled approach to making effective use of any music signals with no, partial, and complete annotations.

One solution is to take the autoencoding approach based on a cyclic architecture consisting of deep discriminative and generative models. Specifically, a (DNN-based) discriminative model called the *encoder* is used for inferring latent variables (musical elements) from observed variables (acoustic features). A (DNN-based) generative model called the *decoder* is then used for reconstructing the observed variables given the latent variables. The encoder and the decoder can be trained jointly in an unsupervised manner, where the

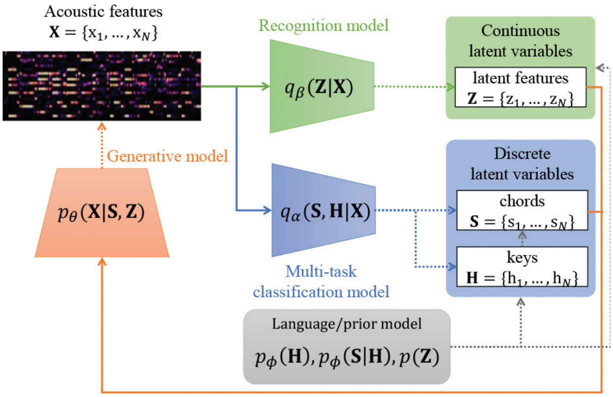

Acoustic features
$\mathbf{X} = \{x_1, \ldots, x_N\}$
Recognition model
$q_\beta(\mathbf{Z}|\mathbf{X})$
Continuous latent variables
latent features
$\mathbf{Z} = \{z_1, \ldots, z_N\}$
Generative model
$p_\theta(\mathbf{X}|\mathbf{S}, \mathbf{Z})$
$q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$
Discrete latent variables
chords
$\mathbf{S} = \{s_1, \ldots, s_N\}$
keys
$\mathbf{H} = \{h_1, \ldots, h_N\}$
Multi-task classification model
Language/prior model
$p_\phi(\mathbf{H}), p_\phi(\mathbf{S}|\mathbf{H}), p(\mathbf{Z})$

Figure 1: The VAE-based multi-task learning approach to joint key and chord estimation, consisting of a multi-task classification model, a recognition model, a language model, and a generative model. The solid arrows indicate data input. The dashed arrows indicate stochastic relationships.

encoder is regularized by the decoder. This framework unifies the the audio-to-label discriminative process and the label-to-audio generative process into a comprehensive model of music understanding and enhances the coherence between the multiple music labels estimated by the encoder and the acoustic features fed into the encoder and predicted by the decoder. Wu *et al.* [34], for example, proposed an automatic chord estimation method based on a variational autoencoder (VAE) [18] that unifies the discriminative and generative models in the framework of amortized variational inference (AVI). This method can significantly improve the performance of chord estimation without increasing the amount of annotated training data.

In this paper, we integrate the VAE-based autoencoding with the multi-task learning in a probabilistic framework to draw the full potential of joint chord and key estimation with a limited amount of training data (Figure 1). More specifically, we formulate a deep *hierarchical* latent variable model to represent the generative process of chroma vectors (observed variables) from discrete key and chord labels and continuous latent features, where the key and chord labels are assumed to follow some language model. In the VAE framework, we introduce a deep classification model that jointly infers chords and keys from chroma vectors, and a deep recognition model that infers latent features from the chroma vectors. All models are then trained jointly, where the generative and language models act as regularizers for the classification model.

The main contribution of this paper is to establish a principled statistical approach to joint chord and key estimation based on the VAE-based framework with multi-task learning. This enables us to deal with completely-, partially-, and non-annotated music recordings in a unified manner with a mixture of supervised, semi-supervised, and unsupervised learning objectives. Another contribution of the paper is to comprehensively investigate parallel, branching, and sequential architectures for the chord and key classification model and autoregressive, Markov, and uniform architectures for the language model.

2 Related Work

This section reviews related works on single- and multi-task music analysis based on machine-learning strategies.

2.1 Generative Approach

Generative modeling is the traditional approach in chord and key estimation tasks. For example, hidden Markov models (HMMs) [31] have widely been used for representing the relationships between a sequence of chords [5, 7, 23] or keys [4] and that of audio features, where the latent states corresponding to musical symbols make a transition at each time step. The sequence of latent states is typically assumed to follow a first-order Markov model, i.e., the transition from a current state to a next state depends on the current state only. In addition, the feature vectors are assumed to be conditionally independent from each other. This simplification enables the optimal state sequence to be analytically inferred from an observed feature sequence with the Viterbi algorithm. On the other hand, the expression capability of the HMM is severely limited by the unrealistic assumptions required for tractable inference.

2.2 Discriminative Approach

DNNs have gained a lot of attention as powerful discriminative models for estimating the posterior probabilities of chords [14, 21, 26, 36] and keys [20, 32] from audio features. In general, a DNN is trained in a supervised manner using annotated music signals such that the posterior probabilities of the annotations conditioned by the audio features are maximized. DNN-based methods use lower-level audio representations as the input and outperform the HMM-based generative methods [19] thanks to the excellent expression capability of the deep architecture.

In chord estimation, a *chord language model* is often integrated with a classification model to yield temporally-coherent chord labels in the inference stage. The language model is typically implemented using an HMM or a linear-chain conditional random field (CRF) [19, 36]. This is similar in form to the DNN-HMM hybrid model for automatic speech recognition (ASR) [12].

Recurrent neural networks (RNNs) have also been used for representing longer-term dependencies of label sequences [21, 33]. The optimal label sequence that maximizes the product of the posterior given by the classification model and the likelihood given by the language model is estimated using the Viterbi or beam-search algorithm. This approach has been successful in filtering out over-frequent transitions that are not supposed to appear.

2.3 Autoencoding Approach

Deep generative and discriminative models have recently been integrated for unsupervised or semi-supervised learning. In the field of ASR, some studies have tried to jointly train a speech-to-text model with a text-to-speech model to improve the performance of ASR by using both annotated and non-annotated speech signals [13]. In the field of MIR, Choi and Cho [8] proposed an unsupervised drum transcription method that trains a deep transcription model such that a spectrogram generated from a transcription result with a synthesizer using drum sound samples is made close to the observed spectrogram.

The VAE has widely been used for integrating deep generative and discriminative models. In chord estimation, Wu *et al.* [34] proposed a deep latent variable model representing the generative process of observed chroma vectors from latent chord labels following a Markov language model. A classification model is then introduced for inferring chord labels from chroma vectors. The generative, classification, and language models are unified to form a VAE that can be trained in a semi-supervised manner. The main difference of the training objective between the VAE and the basic autoencoder lies in the existence of the regularization terms with respect to the priors of latent variables. The Markov language model encourages the classification model to output consistent chord labels.

2.4 Multi-Task Learning Approach

At the heart of multi-task music analysis is representing the semantic relationships between multiple musical elements. In earlier research, HMMs have often been used for modeling the generative process of labels and features. Lee and Slaney [23] simultaneously trained multiple HMMs corresponding to different keys. Given a sequence of chroma features, the optimal chord sequence and musical key can be jointly determined by inferring the optimal sequences for all the HMMs and selecting one with the highest likelihood. In this way, mutually dependent musical elements are hierarchically formulated as the emission probabilities conditioned by the latent states of HMMs [25, 27, 29]. This approach can explicitly reflect our musical knowledge, e.g., that a chord sequence depends on a key sequence [22], and that chord transitions are more likely to occur at beat positions. Papadopoulo and Peeters [28] proposed a joint chord and downbeat estimation method that focuses on the relationships between downbeats and chord boundaries.

As for the discriminative approach, the common multi-task learning method to joint estimation of multiple musical elements is to train a DNN that has branching outputs for predicting the label posteriors from music signals. Considering the mutual dependency of rhythmic musical elements, for example, Böck *et al.* [3] attempted joint estimation of tempos and beats and demonstrated the mutual benefit for beat tracking. They further proposed a joint tempo, beat, and downbeat estimation method [2]. In chord estimation, Mcfee and Bello [26] used a structured training technique that jointly estimates root notes, bass notes, and chord tones as well as chord labels. Chen and Su [6] proposed the harmony transformer for jointly estimating a chord sequence with transition positions. Jiang *et al.* [16] used crowd-sourced data to train a DNN-based multi-task classification model that jointly estimates keys, chords, beats, and melody scales. At the MIREX2019 competition, the multi-task classification method improved the key estimation accuracy.

Our earlier work [37] integrated the multi-task learning with the autoencoding framework for joint chord and key estimation. This method had room for further improvement because the latent variables (chords, keys, and latent features) were treated equally without explicitly considering their hierarchical relationships. In addition, supervised, semi-supervised, and unsupervised conditions had not been fully investigated, i.e., the method cannot make maximum use of ground-truth annotations in a unified multi-task learning framework. In contrast, the VAE-based framework proposed in this paper reflects a hierarchy between keys, chords, and chroma vectors and can be trained with an objective function based on a mixture of all the three conditions. In addition, we investigate various architectures for implementing the classification and language models.

3 Proposed Method

We explain the proposed method of joint chord and key estimation. In our method, the classification and language models can be implemented in various ways. As for the classification model, we could use a **parallel** model that separately infers chord and key labels or a **branching** model that jointly infers both labels. We could also use a **sequential** model that first predicts the chord labels from chroma vectors and then the key labels from the chord labels. To formulate the language model that favors consistent chord and key labels, we use a deep **autoregressive** model implemented with an RNN or a **Markov** language model instead of the basic **uniform** model.

3.1 Generative Model

Let $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ be a sequence of chroma vectors (observed variables) where N is the number of frames and $\mathbf{x}_n \in [0,1]^D$ is a multi-band chroma vector representing the pitch class activations of lower, middle, and higher pitch ranges ($D = 36$). The chroma vectors were calculated from the harmonic-CQT representation of music audio using a DNN-based chroma extractor [35] that was reimplemented and trained on the slakh2100 [24] dataset. We introduce three kinds of latent variables, namely a sequence of chord labels $\mathbf{S} = \{\mathbf{s}_n\}_{n=1}^N$, a sequence of key labels $\mathbf{H} = \{\mathbf{h}_n\}_{n=1}^N$, and a sequence of latent features $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$, where $\mathbf{s}_n \in \{0,1\}^{K_S}$ and $\mathbf{h}_n \in \{0,1\}^{K_H}$ represent the chord and key labels at frame n, respectively, and $\mathbf{z}_n \in \mathbb{R}^L$ is complementary information that represents how $\mathbf{x}_n$ is deviated from a basic chroma pattern specified by the discrete variable $\mathbf{s}_n$ ($L = 64$ in this paper). Let $\mathbf{s}_0$, $\mathbf{h}_0$, and $\mathbf{z}_0$ be dummy symbols at frame 0 representing the beginning of a sequence. The chord vocabulary consists of all possible combinations of 12 root notes with six types of triad chords (abbreviated as *maj, min, aug, dim, sus2, sus4*), and one non-chord label ($K_S = 73$). The key vocabulary consists of *major* and *minor* keys ($K_H = 24$).

Considering a typical process of music composition, we assume that the observation $\mathbf{X}$ is generated by the following procedure (Figure 2):

1. A key progression $\mathbf{H}$ is stochastically determined under a prior distribution $p(\mathbf{H})$.
2. Given $\mathbf{H}$, a chord progression $\mathbf{S}$ is stochastically determined under a conditional prior distribution $p(\mathbf{S}|\mathbf{H})$.
3. A latent feature sequence $\mathbf{Z}$ is generated under a prior distribution $p(\mathbf{Z})$.
4. Given $\mathbf{S}$ and $\mathbf{Z}$, a chroma vector sequence $\mathbf{X}$ is stochastically generated under a conditional distribution $p(\mathbf{X}|\mathbf{S},\mathbf{Z})$.

The joint probability is thus decomposed as follows:

$$p_{\theta,\phi}(\mathbf{X},\mathbf{S},\mathbf{H},\mathbf{Z}) = p_\theta(\mathbf{X}|\mathbf{S},\mathbf{Z})p_\phi(\mathbf{S},\mathbf{H})p(\mathbf{Z}), \tag{1}$$

where $p_\theta(\mathbf{X}|\mathbf{S},\mathbf{Z})$ is a generative model of $\mathbf{X}$ conditioned by both $\mathbf{S}$ and $\mathbf{Z}$, $p_\phi(\mathbf{S},\mathbf{H})$ is a prior of $\mathbf{S}$ and $\mathbf{H}$, i.e., a unified language model that jointly represents $\mathbf{S}$ and $\mathbf{H}$, and $p(\mathbf{Z})$ is a prior of $\mathbf{Z}$. We formulate $p_\theta(\mathbf{X}|\mathbf{S},\mathbf{Z})$ as follows:

$$p_\theta(\mathbf{X}|\mathbf{S},\mathbf{Z}) = \prod_{n=1}^N \prod_{d=1}^D \text{Bernoulli}(x_{nd}|[\boldsymbol{\omega}_\theta(\mathbf{S},\mathbf{Z})]_{nd}), \tag{2}$$

where $\boldsymbol{\omega}_\theta(\mathbf{S},\mathbf{Z}) \in [0,1]^{ND}$ is the output of a DNN with parameters θ and $[\cdot]_i$ and $[\cdot]_{ij}$ denotes the i-th element and the ij-th block, respectively.

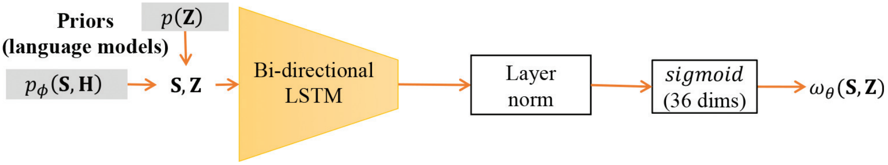


Figure 2: The hierarchical generative model of keys, chords, latent features, and chroma vectors.

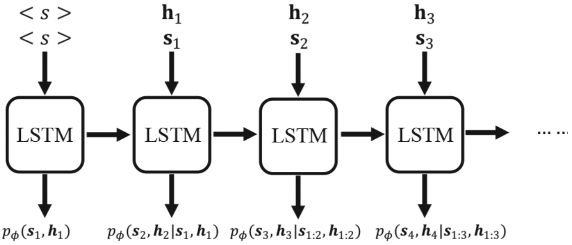

$<s>$
$<s>$
$\mathbf{h}_1$
$\mathbf{s}_1$
$\mathbf{h}_2$
$\mathbf{s}_2$
$\mathbf{h}_3$
$\mathbf{s}_3$
LSTM
LSTM
LSTM
LSTM
……
$p_\phi(\mathbf{s}_1,\mathbf{h}_1)$
$p_\phi(\mathbf{s}_2,\mathbf{h}_2|\mathbf{s}_1,\mathbf{h}_1)$
$p_\phi(\mathbf{s}_3,\mathbf{h}_3|\mathbf{s}_{1:2},\mathbf{h}_{1:2})$
$p_\phi(\mathbf{s}_4,\mathbf{h}_4|\mathbf{s}_{1:3},\mathbf{h}_{1:3})$

Figure 3: The joint language model $p_\phi(\mathbf{S},\mathbf{H})$ with a deep autoregressive architecture. <s> is a special symbol representing the *start of sentence*.

3.2 Language Models

We implement the the language model $p_\phi(\mathbf{S},\mathbf{H})$ of the discrete variables $\mathbf{S}$ and $\mathbf{H}$ with an autoregressive, Markov, or uniform model. In the autoregressive model, $p_\phi(\mathbf{S},\mathbf{H})$ is directly formulated without factorization. In the Markov or uniform model, in contrast, a key language model $p_\phi(\mathbf{H})$ and a chord language model $p_\phi(\mathbf{S}|\mathbf{H})$ conditioned by $\mathbf{H}$ are separately formulated as follows:

$$p_\phi(\mathbf{S},\mathbf{H}) = p_\phi(\mathbf{S}|\mathbf{H})p_\phi(\mathbf{H}). \tag{3}$$

3.2.1 Autoregressive Model

As shown in Figure 3, since the relationships between $\mathbf{S}$ and $\mathbf{H}$ are hard to represent explicitly, we formulate $p_\phi(\mathbf{S},\mathbf{H})$ in an autoregressive manner as follows:

$$p_\phi(\mathbf{S},\mathbf{H}) = \prod_{n=1}^{N} \mathrm{Cat.}(\mathbf{s}_n,\mathbf{h}_n|\boldsymbol{\omega}_\phi(\mathbf{s}_{0:n-1},\mathbf{h}_{0:n-1})), \tag{4}$$

where $i{:}j$ represents a set of indices (integers) from i to j, $\boldsymbol{\omega}_\phi(\mathbf{s}_{0:n-1},\mathbf{h}_{0:n-1}) \in [0,1]^{K_S+K_H}$ is the output of a DNN with parameters ϕ at frame n based on the whole history $(\mathbf{s}_{0:n-1},\mathbf{h}_{0:n-1})$. This model sequentially takes as input $(\mathbf{s}_{n-1},\mathbf{h}_{n-1})$ at frame $n-1$ and predicts $(\mathbf{s}_n,\mathbf{h}_n)$ at frame n in an autoregressive manner.

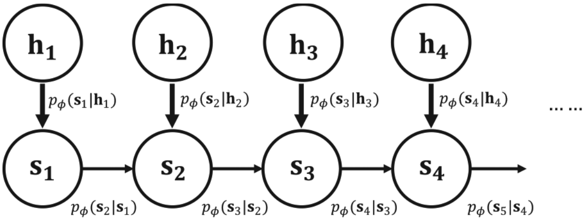

$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3$
$\mathbf{h}_4$
$p_\phi(\mathbf{s}_1|\mathbf{h}_1)$
$p_\phi(\mathbf{s}_2|\mathbf{h}_2)$
$p_\phi(\mathbf{s}_3|\mathbf{h}_3)$
$p_\phi(\mathbf{s}_4|\mathbf{h}_4)$
……
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_3$
$\mathbf{s}_4$
$p_\phi(\mathbf{s}_2|\mathbf{s}_1)$
$p_\phi(\mathbf{s}_3|\mathbf{s}_2)$
$p_\phi(\mathbf{s}_4|\mathbf{s}_3)$
$p_\phi(\mathbf{s}_5|\mathbf{s}_4)$

Figure 4: The conditional language model $p_\phi(\mathbf{S}|\mathbf{H})$ with a key-dependent first-order Markov architecture.

3.2.2 Markov Model

As shown in Figure 4, we formulate a conditional Markov model for implementing $p_\phi(\mathbf{S}|\mathbf{H})$ and $p_\phi(\mathbf{H})$ separately. Specifically, we assume that the current chord label $\mathbf{s}_n$ depends on the previous chord label $\mathbf{s}_{n-1}$ and the current key label $\mathbf{h}_n$ and that the key labels are uniformly distributed as follows:

$$p_\phi(\mathbf{S}|\mathbf{H}) = \prod_{n=1}^{N} \text{Categorical}(\mathbf{s}_n|\boldsymbol{\phi}(\mathbf{h}_n, \mathbf{s}_{n-1})), \tag{5}$$

$$p_\phi(\mathbf{H}) = \prod_{n=1}^{N} \text{Categorical}\big(\mathbf{h}_n|\tfrac{1}{K_H}\mathbf{1}_{K_H}\big), \tag{6}$$

where $\boldsymbol{\phi}(\mathbf{h}_n, \mathbf{s}_{n-1}) \in [0,1]^{K_S}$ is the chord probabilities at frame n conditioned by the previous chord $\mathbf{s}_{n-1}$ and the current key $\mathbf{h}_n$. The parameters $\boldsymbol{\phi}$ are pretrained using a dataset of key and chord sequences.

3.2.3 Uniform Model

The most basic approach is to assume both $\mathbf{S}$ and $\mathbf{H}$ to be uniformly distributed as follows:

$$p_\phi(\mathbf{S}|\mathbf{H}) = \prod_{n=1}^{N} \text{Categorical}\big(\mathbf{s}_n|\tfrac{1}{K_S}\mathbf{1}_{K_S}\big), \tag{7}$$

$$p_\phi(\mathbf{H}) = \prod_{n=1}^{N} \text{Categorical}\big(\mathbf{h}_n|\tfrac{1}{K_H}\mathbf{1}_{K_H}\big), \tag{8}$$

where $\mathbf{1}_K$ is the all-one vector of size L. The joint likelihood of $\mathbf{H}$ and $\mathbf{S}$ is thus constant.

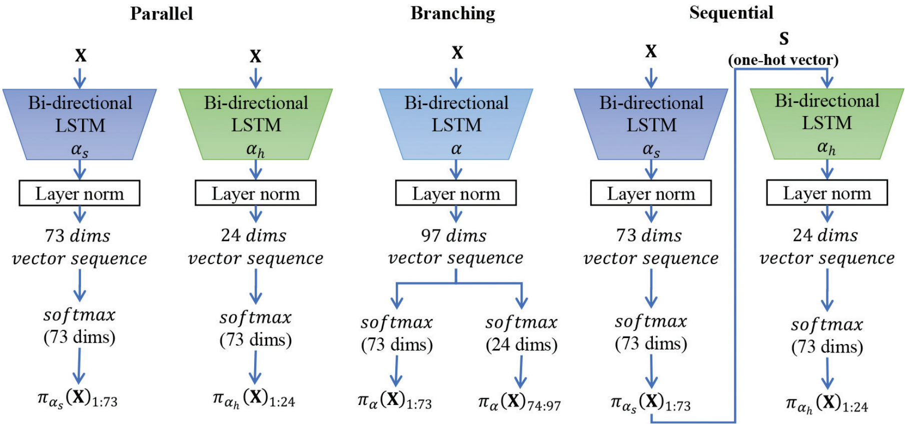

Parallel
Branching
Sequential
X
S
(one-hot vector)
Bi-directional LSTM α_s
Bi-directional LSTM α_h
Bi-directional LSTM α
Layer norm
73 dims vector sequence
24 dims vector sequence
97 dims vector sequence
softmax (73 dims)
softmax (24 dims)
$\pi_{\alpha_s}(\mathbf{X})_{1:73}$
$\pi_{\alpha_h}(\mathbf{X})_{1:24}$
$\pi_{\alpha}(\mathbf{X})_{1:73}$
$\pi_{\alpha}(\mathbf{X})_{74:97}$

Figure 5: The joint chord and key classification models with parallel, branching, and sequential architectures.

3.3 Classification and Recognition Models

Given chroma vectors $\mathbf{X}$ as observed data, we aim to infer the latent variables $\mathbf{S}$, $\mathbf{H}$, and $\mathbf{Z}$. In the framework of AVI, we introduce a variational distribution $q(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X})$ to approximate the true posterior $p(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X})$. As shown in Figure 5, we consider three implementations for the classification model of chords $\mathbf{S}$ and keys $\mathbf{H}$.

3.3.1 Parallel Model

Assuming the conditional independence of the latent variables $\mathbf{S}$, $\mathbf{H}$, and $\mathbf{Z}$ in the posterior space, We decompose the variational posterior as follows:

$$q_{\alpha,\beta}(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X}) = q_{\alpha_s}(\mathbf{S}|\mathbf{X}) q_{\alpha_h}(\mathbf{H}|\mathbf{X}) q_\beta(\mathbf{Z}|\mathbf{X}), \tag{9}$$

where $q_{\alpha_s}(\mathbf{S}|\mathbf{X})$, $q_{\alpha_h}(\mathbf{H}|\mathbf{X})$, and $q_\beta(\mathbf{Z}|\mathbf{X})$ are given by

$$q_{\alpha_s}(\mathbf{S}|\mathbf{X}) = \prod_{n=1}^{N} \text{Categorical}(\mathbf{s}_n | [\boldsymbol{\pi}_{\alpha_s}(\mathbf{X})]_n), \tag{10}$$

$$q_{\alpha_h}(\mathbf{H}|\mathbf{X}) = \prod_{n=1}^{N} \text{Categorical}(\mathbf{h}_n | [\boldsymbol{\pi}_{\alpha_h}(\mathbf{X})]_n), \tag{11}$$

$$q_\beta(\mathbf{Z}|\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{z}_n | [\boldsymbol{\mu}_\beta(\mathbf{X})]_n, [\boldsymbol{\sigma}^2_\beta(\mathbf{X})]_n), \tag{12}$$

where $\boldsymbol{\pi}_{\alpha_s}(\mathbf{X}) \in [0,1]^{NK_S}$ is the output of a DNN with parameters α_s, $\boldsymbol{\pi}_{\alpha_h}(\mathbf{X}) \in [0,1]^{NK_H}$ is that of a DNN with parameters α_h, and $\boldsymbol{\mu}_\beta(\mathbf{X}) \in \mathbb{R}^{NL}$ and $\boldsymbol{\sigma}^2_\beta(\mathbf{X}) \in \mathbb{R}^{NL}_+$ are the outputs of a DNN with parameters β.

3.3.2 Branching Model

To consider the mutual dependency between the chords $\mathbf{S}$ and the keys $\mathbf{H}$, we jointly infer $\mathbf{S}$ and $\mathbf{H}$ from $\mathbf{X}$ as follows

$$q_{\alpha,\beta}(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X}) = q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X}) q_\beta(\mathbf{Z}|\mathbf{X}), \tag{13}$$

where $q_\beta(\mathbf{Z}|\mathbf{X})$ is the same as Equation (12) and $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$ is given by

$$q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X}) = \prod_{n=1}^{N} \mathrm{Categorical}(\mathbf{s}_n, \mathbf{h}_n | [\boldsymbol{\pi}_\alpha(\mathbf{X})]_n), \tag{14}$$

where $\pi_\alpha(\mathbf{X}) \in [0,1]^{NK_S}$ is the output of a DNN with parameters α.

3.3.3 Sequential Model

Based on the reasonable assumption that the keys $\mathbf{H}$ can be determined only from the chords $\mathbf{S}$ in the symbolic domain without referring to the acoustic data $\mathbf{X}$, we formulate a sequential estimation process as follows:

$$q_{\alpha,\beta}(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X}) = q_{\alpha_s}(\mathbf{S}|\mathbf{X}) q_{\alpha_h}(\mathbf{H}|\mathbf{S}) q_\beta(\mathbf{Z}|\mathbf{X}), \tag{15}$$

where $q_{\alpha_s}(\mathbf{S}|\mathbf{X})$ and $q_\beta(\mathbf{Z}|\mathbf{X})$ are the same as Equations (10) and (12), respectively, and $q_{\alpha_h}(\mathbf{H}|\mathbf{S})$ is given by

$$q_{\alpha_h}(\mathbf{H}|\mathbf{S}) = \prod_{n=1}^{N} \mathrm{Categorical}(\mathbf{h}_n | [\boldsymbol{\pi}_{\alpha_h}(\mathbf{S})]_n), \tag{16}$$

where $\boldsymbol{\pi}_{\alpha_h}(\mathbf{S}) \in [0,1]^{NK_H}$ is the output of a DNN with parameters α_h.

3.4 Unsupervised Training

Under an unsupervised condition that only chroma vectors $\mathbf{X}$ are given, we jointly train the generative and classification models such that the log-evidence $\log p_{\theta,\phi}(\mathbf{X})$ is maximized. We use the AVI technique that introduces a DNN-based factorizable variational posterior $q_{\alpha,\beta}(\mathbf{Z}, \mathbf{S}, \mathbf{H}|\mathbf{X}) = q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X}) q_\beta(\mathbf{Z}|\mathbf{X})$ with parameters α and β and maximizes a lower bound $\mathcal{L}_{\mathbf{X}}(\theta, \phi, \alpha, \beta)$ of

$\log p_{\theta,\phi}(\mathbf{X})$, which is given by

$$\begin{aligned}
\log p_{\theta,\phi}(\mathbf{X}) &= \log \iiint p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H}) d\mathbf{Z} d\mathbf{S} d\mathbf{H} \\
&= \log \iiint \frac{q_{\alpha,\beta}(\mathbf{Z},\mathbf{S},\mathbf{H}|\mathbf{X})}{q_{\alpha,\beta}(\mathbf{Z},\mathbf{S},\mathbf{H}|\mathbf{X})} p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H}) d\mathbf{Z} d\mathbf{S} d\mathbf{H} \\
&\geq \iiint q_{\alpha,\beta}(\mathbf{Z},\mathbf{S},\mathbf{H}|\mathbf{X}) \log \frac{p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H})}{q_{\alpha,\beta}(\mathbf{Z},\mathbf{S},\mathbf{H}|\mathbf{X})} d\mathbf{Z} d\mathbf{S} d\mathbf{H} \\
&= \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X}) q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{S})] + \mathbb{E}_{q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})}[\log p_\phi(\mathbf{S},\mathbf{H})] \\
&\quad + \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{Z}) - \log q_\beta(\mathbf{Z}|\mathbf{X})] - \mathbb{E}_{q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})}[\log q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})] \\
&\approx \frac{1}{I} \sum_{i=1}^{I} (\log p_\theta(\mathbf{X}|\mathbf{Z}_i,\mathbf{S}_i) + \log p_\phi(\mathbf{S}_i,\mathbf{H}_i)) \\
&\quad - \mathrm{KL}(q_\beta(\mathbf{Z}|\mathbf{X}) \| p(\mathbf{Z})) + \mathrm{Entropy}[q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})], \\
&\overset{\text{def}}{=} \mathcal{L}_{\mathbf{X}}(\theta,\phi,\alpha,\beta), \qquad (17)
\end{aligned}$$

where $\{\mathbf{Z}_i,\mathbf{S}_i,\mathbf{H}_i\}_{i=1}^{I}$ are a set of I samples drawn from $q_\beta(\mathbf{Z}|\mathbf{X})$ and $q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})$. As in the standard VAE, we set $I = 1$. We found that $(\mathbf{S}_1,\mathbf{H}_1)$ drawn with the Gumbel-softmax trick [15] in a differentiable manner as in Wu *et al.* [34] tend to significantly fluctuate around the maximum-a-posteriori (MAP) estimates of $\mathbf{H}$ and $\mathbf{S}$, denoted by $\mathbf{S}^*$ and $\mathbf{H}^*$:

$$(\mathbf{S}^*,\mathbf{H}^*) = \mathrm{argmax}_{(\mathbf{S},\mathbf{H})} q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X}), \qquad (18)$$

To stabilize the training process and encourage the convergence, we instead use the one-hot vectors $(\mathbf{S}^*,\mathbf{H}^*)$ as $(\mathbf{S}_1,\mathbf{H}_1)$ while making $\mathcal{L}_{\mathbf{X}}(\theta,\phi,\alpha,\beta)$ differentiable with respect to each parameter using the following calculation technique:

$$(\mathbf{s}_n^{soft},\mathbf{h}_n^{soft}) = \mathrm{softmax}(\log[\boldsymbol{\pi}_\alpha(\mathbf{X})]_n), \qquad (19)$$

$$(\mathbf{s}_n^{hard},\mathbf{h}_n^{hard}) = \mathrm{hardmax}(\log[\boldsymbol{\pi}_\alpha(\mathbf{X})]_n), \qquad (20)$$

$$\begin{cases} \mathbf{s}_{1,n} = \mathbf{s}_n^{soft} + \mathbf{s}_n^{hard} - \mathrm{detach}(\mathbf{s}_n^{soft}), \\ \mathbf{h}_{1,n} = \mathbf{h}_n^{soft} + \mathbf{h}_n^{hard} - \mathrm{detach}(\mathbf{h}_n^{soft}), \end{cases} \qquad (21)$$

where $\mathbf{s}_n^{soft}$ and $\mathbf{h}_n^{soft}$ are the posterior probability vectors given by $\boldsymbol{\pi}_\alpha$ at frame n, $\mathbf{s}_n^{hard}$ and $\mathbf{h}_n^{hard}$ are the one-hot vectors that represent the MAP estimates at frame n, and detach($\mathbf{x}$) denotes an operator that detaches a vector $\mathbf{x}$ from the computation graph used for backpropagation. Although hardmax($\mathbf{x}$) is a non-differentiable operator, Equation (21) can yield $\mathbf{S}_1 = \{\mathbf{s}_{1,n}\}_{n=1}^{N}$ and $\mathbf{H}_1 = \{\mathbf{h}_{1,n}\}_{n=1}^{N}$ in a differentiable and deterministic manner.

The maximization of $\mathcal{L}_{\mathbf{X}}(\theta, \phi, \alpha, \beta)$ is equivalent to the minimization of the KL divergence from the variational posterior distribution $q_{\alpha,\beta}(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X})$ to the true posterior distribution $p_{\theta,\phi}(\mathbf{S}, \mathbf{H}, \mathbf{Z}|\mathbf{X})$ [18]. In Equation (17), the entropy of the language model, Entropy$[q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})]$, is computed according to its architecture. In the autoregressive model given by Equation (4), the entropy is approximately computed with Monte Carlo integration using only the sample $(\mathbf{S}_1, \mathbf{H}_1)$. In the Markov model given by Equations (5) and (6), the entropy can be calculated analytically using a dynamic programming technique [34] as follows:

$$\gamma(\mathbf{h}_1) \triangleq \log p_\phi(\mathbf{h}_1), \tag{22}$$

$$\gamma(\mathbf{h}_n) \triangleq \sum_{\mathbf{h}_{n-1}} q_\alpha(\mathbf{h}_{n-1}|\mathbf{X})\big(\gamma(\mathbf{h}_{n-1}) + \log p_\phi(\mathbf{h}_n|\mathbf{s}_{n-1})\big), \tag{23}$$

$$\mathbb{E}_{q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})}[\log p_\phi(\mathbf{H})] = \sum_{\mathbf{h}_N} q_\alpha(\mathbf{h}_N|\mathbf{X})\gamma(\mathbf{h}_N). \tag{24}$$

Similarly, the expectation term for $p_\phi(\mathbf{S}|\mathbf{H})$ is given by

$$\gamma(\mathbf{s}_1) \triangleq \log p_\phi(\mathbf{s}_1), \tag{25}$$

$$\begin{aligned}\gamma(\mathbf{s}_n) \triangleq & \sum_{\mathbf{s}_{n-1}} \sum_{\mathbf{h}_{n-1}} q_\alpha(\mathbf{s}_{n-1}, \mathbf{h}_{n-1}|\mathbf{X}) \\ & \times \big(\gamma(\mathbf{s}_{n-1}) + \log p_\phi(\mathbf{s}_n|\mathbf{s}_{n-1}, \mathbf{h}_{n-1})\big),\end{aligned} \tag{26}$$

$$\mathbb{E}_{q_\alpha(\mathbf{S},\mathbf{H}|\mathbf{X})}[\log p_\phi(\mathbf{S}|\mathbf{H})] = \sum_{\mathbf{s}_N} q_\alpha(\mathbf{s}_N|\mathbf{X})\gamma(\mathbf{s}_N). \tag{27}$$

In the uniform model, the entropy is irrelevant to the maximization of $\mathcal{L}_{\mathbf{X}}(\theta, \phi, \alpha, \beta)$. The regularization based on the uniform model thus corresponds to the maximization of the entropy of the variational posterior $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$.

3.5 Supervised Training

We define objective functions to be maximized with respect to the parameters of the classification, recognition, and generative models under partly or fully supervised conditions.

Under the *partly supervised* condition that $\mathbf{S}$ is available, we aim to maximize a variational lower bound of the log-likelihood $\log p_{\theta,\phi}(\mathbf{X}, \mathbf{S})$ given by

$$\begin{aligned}\log p_{\theta,\phi}(\mathbf{X}, \mathbf{S}) &= \log \int p_{\theta,\phi}(\mathbf{X}, \mathbf{Z}, \mathbf{S}, \mathbf{H}) d\mathbf{Z} d\mathbf{H} \\ &\geq \int q_{\alpha,\beta}(\mathbf{Z}, \mathbf{H}|\mathbf{X}, \mathbf{S}) \log \frac{p_{\theta,\phi}(\mathbf{X}, \mathbf{Z}, \mathbf{S}, \mathbf{H})}{q_{\alpha,\beta}(\mathbf{Z}, \mathbf{H}|\mathbf{X}, \mathbf{S})} d\mathbf{Z} d\mathbf{H}\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{S})] + \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{Z}) - \log q_\beta(\mathbf{Z}|\mathbf{X})] \\
&\quad + \mathbb{E}_{q_{\alpha_h}(\mathbf{H}|\mathbf{X})}[\log p_\phi(\mathbf{S},\mathbf{H}) - \log q_{\alpha_h}(\mathbf{H}|\mathbf{X})] \\
&\approx \frac{1}{I}\sum_{i=1}^{I} (\log p_\theta(\mathbf{X}|\mathbf{Z}_i,\mathbf{S}) + \log p_\phi(\mathbf{S},\mathbf{H}_i)) \\
&\quad - \mathrm{KL}(q_\beta(\mathbf{Z}|\mathbf{X}) \| p(\mathbf{Z})) + \mathrm{Entropy}[q_{\alpha_h}(\mathbf{H}|\mathbf{X})] \\
&\overset{\text{def}}{=} \mathcal{L}_{\mathbf{X},\mathbf{S}}(\theta,\phi,\alpha,\beta).
\end{aligned} \tag{28}
$$

When the **sequential** architecture is used, $q_{\alpha_h}(\mathbf{H}|\mathbf{S})$ given by Equation (16) is used instead of $q_{\alpha_h}(\mathbf{H}|\mathbf{X})$. Similarly, when $\mathbf{H}$ is available, we aim to maximize a variational lower bound of the log-likelihood $\log p_{\theta,\phi}(\mathbf{X},\mathbf{H})$ given by

$$
\begin{aligned}
\log p_{\theta,\phi}(\mathbf{X},\mathbf{H}) &= \log \int p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H}) d\mathbf{Z} d\mathbf{S} \\
&\geq \int q_{\alpha,\beta}(\mathbf{Z},\mathbf{S}|\mathbf{X},\mathbf{H}) \log \frac{p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H})}{q_{\alpha,\beta}(\mathbf{Z},\mathbf{S}|\mathbf{X},\mathbf{H})} d\mathbf{Z} d\mathbf{S} \\
&= \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{S})] + \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{Z}) - \log q_\beta(\mathbf{Z}|\mathbf{X})] \\
&\quad + \mathbb{E}_{q_{\alpha_s}(\mathbf{S}|\mathbf{X})}[\log p_\phi(\mathbf{S},\mathbf{H}) - \log q_{\alpha_s}(\mathbf{S}|\mathbf{X})] \\
&\approx \frac{1}{I}\sum_{i=1}^{I} (\log p_\theta(\mathbf{X}|\mathbf{Z}_i,\mathbf{S}_i) + \log p_\phi(\mathbf{S}_i,\mathbf{H})) \\
&\quad - \mathrm{KL}(q_\beta(\mathbf{Z}|\mathbf{X}) \| p(\mathbf{Z})) + \mathrm{Entropy}[q_{\alpha_s}(\mathbf{S}_i|\mathbf{X})] \\
&\overset{\text{def}}{=} \mathcal{L}_{\mathbf{X},\mathbf{H}}(\theta,\phi,\alpha,\beta).
\end{aligned} \tag{29}
$$

Under the *fully supervised* condition that both $\mathbf{S}$ and $\mathbf{H}$ are available, we aim to maximize a variational lower bound of the log-likelihood $\log p_{\theta,\phi}(\mathbf{X},\mathbf{S},\mathbf{H})$ given by

$$
\begin{aligned}
\log p_{\theta,\phi}(\mathbf{X},\mathbf{S},\mathbf{H}) &= \log \int p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H}) d\mathbf{Z} \\
&\geq \int q_\beta(\mathbf{Z}|\mathbf{X},\mathbf{S},\mathbf{H}) \log \frac{p_{\theta,\phi}(\mathbf{X},\mathbf{Z},\mathbf{S},\mathbf{H})}{q_\beta(\mathbf{Z}|\mathbf{X},\mathbf{S},\mathbf{H})} d\mathbf{Z} \\
&= \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{S})] \\
&\quad + \mathbb{E}_{q_\beta(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{Z}) - \log q_\beta(\mathbf{Z}|\mathbf{X})] + \log p_\phi(\mathbf{S},\mathbf{H}) \\
&\approx \frac{1}{I}\sum_{i=1}^{I} \log p_\theta(\mathbf{X}|\mathbf{Z}_i,\mathbf{S}) - \mathrm{KL}(q_\beta(\mathbf{Z}|\mathbf{X}) \| p(\mathbf{Z})) + \log p_\phi(\mathbf{S},\mathbf{H}) \\
&\overset{\text{def}}{=} \mathcal{L}_{\mathbf{X},\mathbf{S},\mathbf{H}}(\theta,\phi,\alpha,\beta).
\end{aligned} \tag{30}
$$

Since the chord and key classification models $q_{\alpha_h}(\mathbf{H}|\mathbf{X})$ $q_{\alpha_s}(\mathbf{S}|\mathbf{X})$, and $q_{\alpha}(\mathbf{S},\mathbf{H}|\mathbf{X})$, which are the main optimization targets, are not involved in the lower bounds, the posterior probability terms are added as follows:

$$\mathcal{L}'_{\mathbf{X},\mathbf{S}}(\theta,\phi,\alpha,\beta) = \mathcal{L}_{\mathbf{X},\mathbf{S}}(\theta,\phi,\alpha,\beta) + \log q_{\alpha_s}(\mathbf{S}|\mathbf{X}), \tag{31}$$

$$\mathcal{L}'_{\mathbf{X},\mathbf{H}}(\theta,\phi,\alpha,\beta) = \mathcal{L}_{\mathbf{X},\mathbf{H}}(\theta,\phi,\alpha,\beta) + \log q_{\alpha_h}(\mathbf{H}|\mathbf{X}), \tag{32}$$

$$\mathcal{L}'_{\mathbf{X},\mathbf{S},\mathbf{H}}(\theta,\phi,\alpha,\beta) = \mathcal{L}_{\mathbf{X},\mathbf{S},\mathbf{H}}(\theta,\phi,\alpha,\beta) + \log q_{\alpha}(\mathbf{S},\mathbf{H}|\mathbf{X}). \tag{33}$$

As in Equation (28), $q_{\alpha_h}(\mathbf{H}|\mathbf{X})$ is replaced with $q_{\alpha_h}(\mathbf{H}|\mathbf{S})$ when the sequential architecture is used. Given chroma vectors $\mathbf{X}$ with or without the ground-truth chords $\mathbf{S}$ and keys $\mathbf{H}$, the total objective function to be maximized is given by

$$\begin{aligned}\mathcal{L}(\theta,\phi,\alpha,\beta) = \sum_{\mathbf{X}} \mathcal{L}_{\mathbf{X}}(\theta,\phi,\alpha,\beta) + \sum_{\mathbf{X},\mathbf{S}} \mathcal{L}'_{\mathbf{X},\mathbf{S}}(\theta,\phi,\alpha,\beta)\\ + \sum_{\mathbf{X},\mathbf{H}} \mathcal{L}'_{\mathbf{X},\mathbf{H}}(\theta,\phi,\alpha,\beta) + \sum_{\mathbf{X},\mathbf{S},\mathbf{H}} \mathcal{L}'_{\mathbf{X},\mathbf{S},\mathbf{H}}(\theta,\phi,\alpha,\beta).\end{aligned} \tag{34}$$

To stabilize the semi-supervised training, we use a curriculum learning technique. First, the parameters are optimized using completely annotated data such that the fourth term of Equation (34) is maximized. In this step, the classification model $q_{\alpha}(\mathbf{S},\mathbf{H}|\mathbf{X})$ is trained solely in a supervised manner, whereas the generative model $p_{\theta}(\mathbf{X}|\mathbf{Z},\mathbf{S})$ and the recognition model $q_{\beta}(\mathbf{Z}|\mathbf{X})$ are jointly trained in an unsupervised manner as in the standard VAE. Then, the parameters are further optimized such that the total objective function $\mathcal{L}(\theta,\phi,\alpha,\beta)$ is maximized.

We use $\mathcal{L}(\theta,\phi,\alpha,\beta)$ given by Equation (34) as an objective function regardless of the *actual* availability of key and/or chord annotations, as proposed in Wu *et al.* [34]. Even under the fully supervised condition that both $\mathbf{S}$ and $\mathbf{H}$ are given, we *simulate* the unsupervised and partly supervised conditions. If one maximizes the fourth term of Equation (34) corresponding to the fully supervised condition, the generative model $p_{\theta}(\mathbf{X}|\mathbf{Z},\mathbf{S})$ is optimized using the ground-truth values of $\mathbf{S}$. In contrast, we optimize $p_{\theta}(\mathbf{X}|\mathbf{Z},\mathbf{S})$ using the estimated values of $\mathbf{S}$ sampled from the chord classification model $q_{\alpha_s}(\mathbf{S}|\mathbf{X})$ as well as the ground-truth values of $\mathbf{S}$. This technique improves the robustness of $p_{\theta}(\mathbf{X}|\mathbf{Z},\mathbf{S})$.

3.6 Prediction

Given the chroma vectors $\mathbf{X}$, the chords and keys are finally obtained with the standard Viterbi decoding on the label posteriors in a post-processing step. Considering the temporal continuity, the self-transition probabilities are set to 0.9 for chords, as suggested in Wu *et al.* [34]. Since key labels make much fewer transitions, we heuristically set the self-transition probabilities for keys to 0.95.

4 Evaluation

We report comparative experiments conducted for evaluating the effectiveness of the proposed method.

4.1 Experimental Conditions

We explain the datasets, compared methods, network configurations, and measures used for evaluation.

4.1.1 Datasets

We conducted five-fold cross-validation on 245 songs consisting of 213 songs selected from the Isophonics dataset (220 songs) [11] and the 32 songs selected from the Robbie Williams dataset (63 songs) [9] given time-aligned chord and key annotations, where 38 songs have keys other than major and minor keys (e.g., Mixolydian) and were not used for evaluation.

To investigate the generalization capability, we also used the McGill Billboard dataset [**Burgoyne2011**] for evaluation. Although 186 songs were originally used for evaluation [20],[1] we could not collect the audio recordings of 49 songs, i.e., the remaining 137 songs were used for evaluation in our experiment.[2] More specifically, in each fold of the cross-validation, an ACE method was trained on the training set (196 songs) and then evaluated on not only the test set (49 songs) but also the test set of the McGill Billboard dataset (137 songs). To compensate for the imbalance of key labels, we augmented the training set by pitch-shifting the chroma vectors and the corresponding chord and key labels between one and twelve semitones.

4.1.2 Methods

We tested all nine combinations of the three types of the classification model $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$:

- **Parallel (PR)**: $\mathbf{S}$ and $\mathbf{H}$ are estimated independently from $\mathbf{X}$ with Equations (10) and (11).
- **Branching (BR)**: $\mathbf{S}$ and $\mathbf{H}$ are estimated simultaneously from $\mathbf{X}$ with Equation (14).
- **Sequential (SQ)**: $\mathbf{S}$ and $\mathbf{H}$ are estimated sequentially in this order from $\mathbf{X}$ with Equation (15).

[1] http://www.cp.jku.at/people/korzeniowski/bb.zip
[2] http://sap.ist.i.kyoto-u.ac.jp/members/wu/apsipa2022.txt

and the three types of language model $p_\phi(\mathbf{S}, \mathbf{H})$:

- **Autoregressive (AR)**: $\mathbf{S}$ and $\mathbf{H}$ are jointly represented with a deep autoregressive model given by Equation (4).
- **Markov (MK)**: $\mathbf{S}$ is represented with a Markov model given by Equation (5) conditioned by uniformly distributed $\mathbf{H}$ given by Eq. (6).
- **Uniform (UN)**: $\mathbf{S}$ and $\mathbf{H}$ are uniformly distributed with Equations (7) and (8).

The classification model $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$ was trained with or without the proposed VAE-based regularization. In the non-regularized training, $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$ was optimized such that the fourth term of Equation (34) corresponding to the standard supervised condition was maximized (denoted by **PR**, **BR**, or **SQ**). In the regularized training, $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$ was optimized such that Equation (34) considering the unsupervised and fully and partly supervised conditions was maximized, where the language model $p_\phi(\mathbf{S}, \mathbf{H})$ and the generative model $p_\theta(\mathbf{X}|\mathbf{Z}, \mathbf{S})$ were used for evaluating $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$ in terms of musical naturalness and reconstruction quality, respectively (denoted by {**PR, BR, SQ**}-{**AR, MK, UN**}).

As baselines, we used popular DNN-based chord and key estimation methods [19, 20] (denoted by **MM-C** and **MM-K**) based on convolutional neural networks (CNNs) trained in a fully supervised manner. The pre-trained models of the baseline methods were provided by the *madmom* library [**Bock2016madmom**]. The data used for training the baseline methods were partly different from those used for training our methods, but did not overlap with the test set of the Billboard dataset. The performance comparison of the pre-trained baseline methods with the proposed methods could be considered to be moderately fair.

4.1.3 Configurations

The chord and key classification model $q_\alpha(\mathbf{S}, \mathbf{H}|\mathbf{X})$, the latent feature recognition model $q_\beta(\mathbf{Z}|\mathbf{X})$, and the chroma vector generative model $p_\theta(\mathbf{X}|\mathbf{S}, \mathbf{H}, \mathbf{Z})$ were each implemented with a three-layered bi-directional long short-term memory (BLSTM) network [10] followed by layer normalization [1], where each layer had 128 hidden units in each direction. The output was then transformed into the desired shape using a fully-connected layer, and normalized with the softmax function. The autoregressive language model was implemented with a single-layer uni-directional LSTM network, where each layer had 32 hidden units. The parameters θ, ϕ, α, and β were optimized with Adam [17] with an initial learning rate of 0.001. Each minibatch consisted of 32 sequences randomly picked from training data, where each sequence consisted of 431 frames (20 seconds).

4.1.4 Measures

We measured the frame-level matching rates between the estimated and ground-truth chord and key labels for each piece. To evaluate the discriminative power of the key classification model, we also measured the rates of typical three types of key estimation errors [20] listed below:

- **Parallel**: The estimated and reference keys have the same tonic with different types (e.g., C major and C minor).
- **Relative**: The estimated and reference keys consist of the same set of pitch classes (e.g., C major and A minor).
- **Fifth**: The estimated key is a perfect-5th above the reference key.

In the cross-validation experiment, the overall chord and key matching rates and the overall key error rates were given by averaging the piece-wise rates over the 287 songs. In the cross-dataset experiment, the overall rates were given by averaging the piece-wise rates over the 137 songs and the five folds, where the piece-wise key matching rate was measured by comparing the ground-truth global key with the most frequent key in the estimated keys.

4.2 Experimental Results

We discuss the results of the cross-validation and cross-dataset experiments listed in Tables 1 and 2, respectively.

4.2.1 Non-regularized Training

The top three rows in Table 1 show the performance of the non-regularized classification models in the cross-validation experiment. In key estimation, the **BR** method performed best, followed by the **SQ** and **PR** methods in this order. The parallel and relative key errors were significantly reduced by more than 1 pts. This indicates the effectiveness of the multi-task learning strategy to prevent the estimated keys from being incompatible with the estimated chords. The **SQ** method underperformed the **BR** method in term of key accuracy, but was more effective in reducing the parallel and relative key errors.

The results of the cross-dataset experiment (Table 2) also showed the advantage of the multi-task architectures for key estimation. The **SQ** method achieved significantly better key estimation than the **PR** and **BR** methods and outperformed the baseline **MM-K** method. The **SQ** method reduced all the three typical types of key errors, whereas the **BR** method failed to reduce the relative key errors.

The advantage of the sequential architecture over the branching architecture in key estimation appeared in the difference of the confusion matrices shown

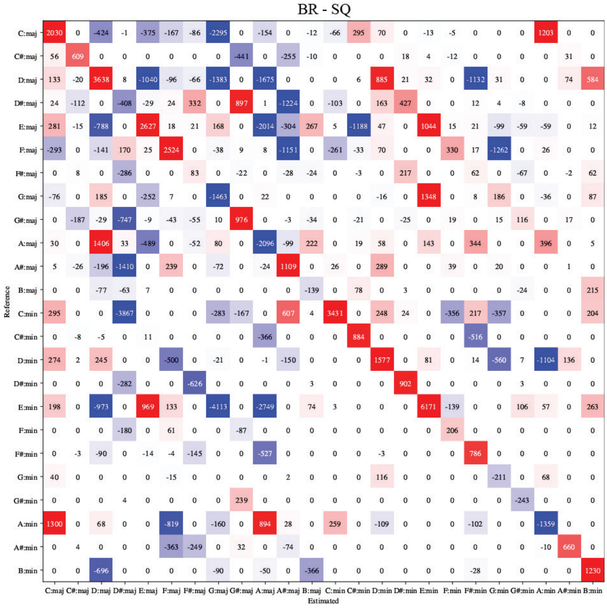
Table 1: Results of cross-validation experiment.

	Accuracy [%]		Key estimation errors [%]		
Method	Chord	Key	Parallel	Relative	Fifth
PR	81.41	76.93	4.79	6.59	4.38
BR	81.14	**81.75**	3.04	4.98	4.86
SQ	**81.51**	78.53	2.48	4.71	4.69
PR-UN	81.95	80.43	4.83	5.36	3.78
PR-MK	82.55	79.17	5.08	6.17	3.24
PR-AR	81.66	80.69	3.93	5.04	4.27
BR-UN	82.28	80.77	3.69	6.13	3.80
BR-MK	82.72	81.29	3.52	5.25	4.37
BR-AR	82.01	81.79	3.29	4.30	5.31
SQ-UN	**82.83**	**84.07**	2.12	4.89	3.12
SQ-MK	82.79	82.68	2.01	5.55	3.17
SQ-AR	82.29	82.77	2.42	5.40	4.28

Table 2: Results of cross-dataset experiment.

	Accuracy [%]		Key estimation errors [%]		
Method	Chord	Key	Parallel	Relative	Fifth
PR	73.43	72.84	3.80	6.71	5.70
BR	72.61	74.45	2.63	7.59	4.09
SQ	**73.98**	**80.19**	2.63	3.80	2.48
PR-UN	75.12	77.81	3.50	5.26	3.80
PR-MK	75.26	79.99	3.50	4.96	2.92
PR-AR	74.39	78.68	2.92	5.69	3.80
BR-UN	74.79	80.87	2.77	4.82	1.90
BR-MK	75.15	80.73	3.50	3.35	3.07
BR-AR	74.15	**81.02**	3.21	3.65	2.77
SQ-UN	**75.37**	80.87	3.07	4.09	2.48
SQ-MK	75.10	80.58	2.92	4.23	3.36
SQ-AR	74.54	80.87	3.65	3.65	2.92
MM-C[19]	77.71	/	/	/	/
MM-K[20]	/	79.56	5.11	5.11	1.46

in Figure 6. Most diagonal elements of the difference matrix were positive, i.e., the **BR** method basically made more accurate predictions than the **SQ** method. In particular, it was significantly less likely to misclassify minor keys



	C:maj	C#:maj	D:maj	D#:maj	E:maj	F:maj	F#:maj	G:maj	G#:maj	A:maj	A#:maj	B:maj	C:min	C#:min	D:min	D#:min	E:min	F:min	F#:min	G:min	G#:min	A:min	A#:min	B:min
C:maj	2030	0	-424	-1	-375	-167	-86	-2295	0	-154	0	-12	-66	295	70	0	-13	-5	0	0	0	1203	0	0
C#:maj	56	609	0	0	0	0	0	0	-441	0	-255	-10	0	0	0	18	4	-12	0	0	0	0	31	0
D:maj	133	-20	3638	8	-1040	-96	-66	-1383	0	-1675	0	0	0	6	885	21	32	0	-1132	31	0	0	74	584
D#:maj	24	-112	0	-408	-29	24	332	0	897	1	-1224	0	-103	0	163	427	0	0	12	4	-8	0	0	0
E:maj	281	-15	-788	0	2627	18	21	168	0	-2014	-304	267	5	-1188	47	0	1044	15	21	-99	-59	-59	0	12
F:maj	-293	0	-141	170	25	2524	0	-38	9	8	-1151	0	-261	-33	70	0	0	330	17	-1262	0	26	0	0
F#:maj	0	8	0	-286	0	0	83	0	-22	0	-28	-24	0	-3	0	217	0	0	62	0	-67	0	-2	62
G:maj	-76	0	185	0	-252	7	0	-1463	0	22	0	0	0	0	-16	0	1348	0	8	186	0	-36	0	87
G#:maj	0	-187	-29	-747	-9	-43	-55	10	976	0	-3	-34	0	-21	0	-25	0	19	0	15	116	0	17	0
A:maj	30	0	1406	33	-489	0	-52	80	0	-2096	-99	222	0	19	58	0	143	0	344	0	0	396	0	5
A#:maj	5	-26	-196	-1410	0	239	0	-72	0	-24	1109	0	26	0	289	0	0	39	0	20	0	0	1	0
B:maj	0	0	-77	-63	7	0	0	0	0	0	0	-139	0	78	0	3	0	0	0	0	-24	0	0	215
C:min	295	0	0	-3867	0	0	0	-283	-167	0	607	4	3431	0	248	24	0	-356	217	-357	0	0	0	204
C#:min	0	-8	-5	0	11	0	0	0	0	-366	0	0	0	884	0	0	0	0	-516	0	0	0	0	0
D:min	274	2	245	0	0	-500	0	-21	0	-1	-150	0	0	0	1577	0	81	0	14	-560	7	-1104	136	0
D#:min	0	0	0	-282	0	0	-626	0	0	0	0	3	0	0	0	902	0	0	0	0	3	0	0	0
E:min	198	0	-973	0	969	133	0	-4113	0	-2749	0	74	3	0	0	0	6171	-139	0	0	106	57	0	263
F:min	0	0	0	-180	0	61	0	0	-87	0	0	0	0	0	0	0	0	206	0	0	0	0	0	0
F#:min	0	-3	-90	0	-14	-4	-145	0	0	-527	0	0	0	0	-3	0	0	0	786	0	0	0	0	0
G:min	40	0	0	0	0	-15	0	0	0	0	2	0	0	0	116	0	0	0	0	-211	0	68	0	0
G#:min	0	0	0	4	0	0	0	0	239	0	0	0	0	0	0	0	0	0	0	0	-243	0	0	0
A:min	1300	0	68	0	0	-819	0	-160	0	894	28	0	259	0	-109	0	0	0	-102	0	0	-1359	0	0
A#:min	0	4	0	0	0	-363	-249	0	32	0	-74	0	0	0	0	0	0	0	0	0	0	-10	660	0
B:min	0	0	-696	0	0	0	0	-90	0	-50	0	-366	0	0	0	0	0	0	-28	0	0	0	0	1230

Figure 6: Difference between the confusion matrices of key estimation by **BR** and **SQ**. Numbers in the matrix represent the number of frames.

as their relative keys. This remarkably improved the accuracy of estimating C minor and E minor keys. On the other hand, the **SQ** method tended to misclassify major keys as their relative and parallel keys, and minor keys as their parallel keys. These errors were the main factor that degraded the accuracy of estimating G major and A minor keys. Besides the typical error types, the **BR** method was less likely to misclassify major keys as other major keys and minor keys as major keys, while it was more likely to misclassify major and minor keys as other minor keys.

The **PR**, **BR**, and **SQ** methods attained almost the same accuracy of chord estimation in the cross-validation and cross-dataset experiments, possibly because chroma vectors were used as input in common. Whereas the **BR** method was beneficial for key estimation thanks to the joint estimation strategy, it had little benefit for chord estimation.

4.2.2 Regularized Training

The bottom nine rows of Table 1 show the performance of the regularization methods in the cross-validation experiment. The **PR-UN** and **BR-UN** methods outperformed the non-regularized **PR** and **BR** methods and the **PR-**

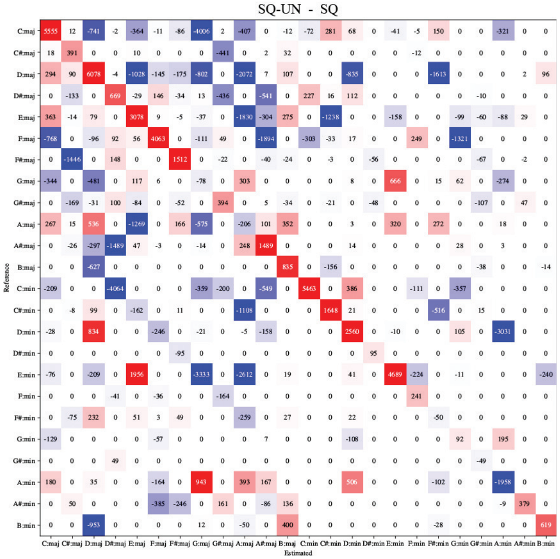


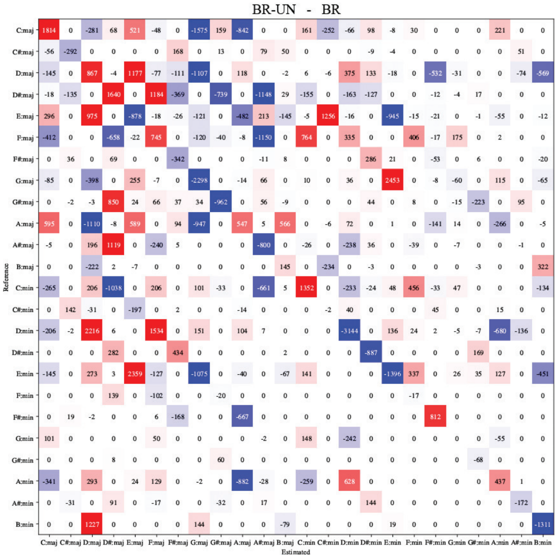
Figure 7: The difference matrix computed from the key confusion matrices obtained by **SQ-UN** and **SQ**.

MK and **BR-MK** methods. As for the sequential classification model, **SQ-UN** performed best in chord estimation, even outperforming the other classification models regularized by the Markov language model. For all classification models, the autoregressive language model showed little performance gain over the other models.

The key estimation accuracy of the sequential classification model was significantly improved with the regularization mechanism (**SQ-UN**, **SQ-MK**, or **SQ-AR**). As seen in chord estimation, the **SQ-UN** method with uniform prior performed better for key estimation, and the **SQ-MK** method with the Markov prior achieved a lower overall performance. More specifically, the regularized classification model made fewer parallel key and fifth key errors, but did not reduce the relative key errors.

In the cross-dataset experiment, the regularized classification models outperformed their non-regularized counterparts. Among the regularized methods, the **BR-AR** method performed best in key estimation and the **SQ-UN** method performed best in chord estimation. Under the cross-dataset setting, the overall key estimation accuracies of the multi-task architectures (**BR-**** and **SQ-****) were higher than the single-task architecture (**PR-****) and the different language models worked comparably as regularizers.

Comparing the key estimation accuracies of the **SQ** and **SQ-UN** methods, we validated the regularization mechanism with the uniform priors. As shown



	C:maj	C#:maj	D:maj	D#:maj	E:maj	F:maj	F#:maj	G:maj	G#:maj	A:maj	A#:maj	B:maj	C:min	C#:min	D:min	D#:min	E:min	F:min	F#:min	G:min	G#:min	A:min	A#:min	B:min
C:maj	1814	0	-281	68	521	-48	0	-1575	159	-842	0	0	161	-252	-66	98	-8	30	0	0	0	221	0	0
C#:maj	-56	-292	0	0	0	0	168	0	13	0	79	50	0	0	0	-9	-4	0	0	0	0	0	51	0
D:maj	-145	0	867	-4	1177	-77	-111	-1107	0	118	0	-2	6	-6	375	133	-18	0	-532	-31	0	0	-74	-569
D#:maj	-18	-135	0	1640	0	1184	-369	0	-739	0	-1148	29	-155	0	-163	-127	0	0	-12	-4	17	0	0	0
E:maj	296	0	975	0	-878	-18	-26	-121	0	-482	213	-145	-5	1256	-16	0	-945	-15	-21	0	-1	-55	0	-12
F:maj	-412	0	0	-658	-22	745	0	-120	-40	-8	-1150	0	764	0	335	0	0	406	-17	175	0	2	0	0
F#:maj	0	36	0	69	0	0	-342	0	0	0	-11	8	0	0	0	286	21	0	-53	0	6	0	0	-20
G:maj	-85	0	-398	0	255	-7	0	-2298	0	-14	66	0	10	0	36	0	2453	0	-8	-60	0	115	0	-65
G#:maj	0	-2	-3	850	24	66	37	34	-962	0	56	-9	0	0	0	44	0	8	0	-15	-223	0	95	0
A:maj	595	0	-1110	-8	589	0	94	-947	0	547	5	566	0	-6	72	0	1	0	-141	14	0	-266	0	-5
A#:maj	-5	0	196	1119	0	-240	5	0	0	0	-800	0	-26	0	-238	36	0	-39	0	-7	0	0	-1	0
B:maj	0	0	-222	2	-7	0	0	0	0	0	0	145	0	-234	0	-3	0	0	0	0	-3	0	0	322
C:min	-265	0	206	-1038	0	206	0	101	-33	0	-661	5	1352	0	-233	-24	48	456	-33	47	0	0	0	-134
C#:min	0	142	-31	0	-197	0	2	0	0	-14	0	0	0	-2	40	0	0	0	45	0	0	15	0	0
D:min	-206	-2	2216	6	0	1534	0	151	0	104	7	0	0	0	-3144	0	136	24	2	-5	-7	-680	-136	0
D#:min	0	0	0	282	0	0	434	0	0	0	0	2	0	0	0	-887	0	0	0	0	169	0	0	0
E:min	-145	0	273	3	2359	-127	0	-1075	0	-40	0	-67	141	0	0	0	-1396	337	0	26	35	127	0	-451
F:min	0	0	0	139	0	-102	0	0	-20	0	0	0	0	0	0	0	0	-17	0	0	0	0	0	0
F#:min	0	19	-2	0	0	6	-168	0	0	-667	0	0	0	0	0	0	0	0	812	0	0	0	0	0
G:min	101	0	0	0	0	50	0	0	0	0	-2	0	148	0	-242	0	0	0	0	0	0	-55	0	0
G#:min	0	0	0	8	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	-68	0	0	0
A:min	-341	0	293	0	24	129	0	-2	0	-882	-28	0	-259	0	628	0	0	0	0	0	0	437	1	0
A#:min	0	-31	0	91	0	-17	0	0	-32	0	17	0	0	0	0	144	0	0	0	0	0	0	-172	0
B:min	0	0	1227	0	0	0	0	144	0	0	0	-79	0	0	0	0	19	0	0	0	0	0	0	-1311

Figure 8: The difference matrix computed from the key confusion matrices obtained by **BR-UN** and **BR**.

in Figure 7, the accuracies for C and E minor keys were improved mainly because these keys were less likely to be misclassified to their relative keys, i.e., D# and G major keys. Similarly, the accuracies for C, D, and F# major keys were improved because of a reduced number of fifth key errors. Besides the relative and fifth key errors, the **SQ-UN** method reduced the errors where the estimated keys were four degrees above the correct keys (e.g., D major to G major), which also significantly contributed to the performance improvement. The regularization, however, slightly increased parallel key errors. We found that the **SQ-UN** method made more parallel key errors than the **SQ** method for D, E, A, and B minor keys. The accuracy for A minor key was degraded by the confusion with G or A major key.

As for the branching classification model, the regularization mechanism made little improvement of key estimation regardless of the language model used for regularization. As seen for the **SQ-UN** method, the **BR-UN** method also tended to make parallel and relative key erorrs. As shown in Figure 8, the accuracies of D and E minor keys were significantly degraded because these keys were often misclassified to their parallel keys, i.e., D and E major keys. In addition, E and G major keys were tended to be misclassified to their relative keys, i.e., C# and E minor keys. Although the **BR-UN** increased and decreased the errors other than the three types, the influence of these errors on the overall performance was comparatively small.

Although the VAE methods also improved chord estimation accuracy, its relationships with the choice of classification and language models differed from those in key estimation. In Table 1, **PR-MK** achieved the highest chord estimation accuracy, and **BR-MK** also performed the best in chord estimation among the **BR-**** methods. In contrast, the methods that used the deep autoregressive language model (****-AR**) had comparatively lower chord estimation accuracies. This difference indicates that for the chord estimation task, the Markov language model is more beneficial for regularizing the classification models than the autoregressive language model. In contrast to key estimation, the proposed methods underperformed the baseline method **MM-C** in chord estimation. A possible reason lies in the difference in the amount of training data, since the **MM-C** method provided by the *madmom* library were trained on a larger dataset. Even though the proposed methods were able to improve chord estimation performance compared to the fully-supervised approach,the performance is still affected by the amount of annotated data.

5 Conclusion

This paper described a DNN-based joint chord and key estimation method that integrates multi-task learning [37] with VAE-based regularized training [34]. We formulated a hierarchical generative model of keys, chords, latent features, and observed chroma vectors with a music language model acting as a prior distribution of the chords and keys. Using the framework of AVI, we then introduced a joint chord and key classification model and a recognition model for inferring latent features. Even when the ground-truth chord and keys are available, all these models are jointly trained such that the sum of the four log-likelihoods corresponding to unsupervised, partly-supervised, and fully-supervised conditions (neither chords nor keys are given, either chords or key are given, both chords and keys are given).

We compared the parallel, branching, and sequential architectures for the classification model and the autoregressive, Markov, and uniform architectures for the language model, with the cross-validation and cross-dataset experiments. We found that the VAE-based regularization and multi-task learning methods improved the chord and key estimation accuracies in both experiments, compared to the non-regularized, single-task methods. This reveals the effectiveness of the proposed VAE-based regularized multi-task learning. Although the VAE-based regularization improved the overall key estimation accuracy, it tended to make more parallel and relative key errors because of the limited capacity of chroma vectors. Since the autoregressive and Markov language model formulations did not show obvious advantage over the naive uniform formulation in the experiments, the effective use of language models in the regularization mechanism remains an open problem. We also plan to extend the hierarchical generative model that regards chroma vectors as latent features and formulate a three-layered VAE, which incorporates a raw audio spectrogram as an observed variable on top of the proposed hierarchical VAE. In this way, the chroma extractor can be trained jointly with the generative and classification models, so that it could provide more suitable acoustic features for chord and key classification.

Biographies

Yiming Wu received the the B.S. and M.S. degrees from Fudan University, Shanghai, China, in 2015 and 2018 respectively and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2021. His main research interest is music content analysis.

Kazuyoshi Yoshii received the M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2005 and 2008, respectively. He is an Associate Professor at the Graduate School of Informatics, Kyoto University, and concurrently the Leader of the Sound Scene Understanding Team, Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo, Japan. His research interests include music information processing, audio signal processing, and statistical machine learning.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[2] S. Böck and M. Davies, “Deconstruct, Analyse, Reconstruct: How to Improve Tempo, Beat, and Downbeat Estimation,” in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Montréal, Canada, 2020, 574–82.

[3] S. Böck, M. E. Davies, and P. Knees, “Multi-Task Learning for Tempo and Beat: Learning One to Improve the Other,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, Delft, Netherlands, 2019, 486–93.

[4] W. Chai and B. Vercoe, “Detection of Key Change in Classical Piano Music,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2005, 468–73.

[5] R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia, “Chord recognition using duration-explicit hidden Markov models,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, Porto, Portugal, 2012, 445–50.

[6] T.-P. Chen and L. Su, "Harmony Transformer: Incorporating Chord Segmentation into Harmony Recognition," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Delft, Netherlands, 2019, 259–67.

[7] T. Cho, "Improved techniques for automatic chord recognition from music audio signals," *PhD thesis*, New York University, 2014.

[8] K. Choi and K. Cho, "Deep unsupervised drum transcription," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Delft, Netherland, 2019, 183–91.

[9] B. D. Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *nDS '13; Proceedings of the 8th International Workshop on Multidimensional Systems*, 2013, 1–6.

[10] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, December 2013, 273–8.

[11] C. Harte, "Towards automatic extraction of harmony information from music signals," *PhD thesis*, Queen Mary University of London, 2010.

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

[13] T. Hori, R. F. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. L. Roux, "Cycle-consistency training for end-to-end speech recognition," in *proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, 6271–5.

[14] E. J. Humphrey and J. P. Bello, "Rethinking Automatic Chord Recognition with Convolutional Neural Networks," in *2012 11th International Conference on Machine Learning and Applications*, Vol. 2, 2012, 357–62.

[15] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," in *proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[16] J. Jiang, G. Xia, and D. B. Carlton, "Mirex 2019 submission: Crowd annotation for audio key estimation," *Abstract of MIREX*, 2019.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, 1–15.

[18] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *proceedings of the International Conference on Learning Representations (ICLR)*, 2014, 1–14.

[19] F. Korzeniowski and G. Widmer, "A fully convolutional deep auditory model for musical chord recognition," in *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016, 13–6.

[20] F. Korzeniowski and G. Widmer, "Genre-Agnostic Key Classification with Convolutional Neural Networks," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, Paris, France, 2018, 264–70.

[21] F. Korzeniowski and G. Widmer, "Improved Chord Recognition by Combining Duration and Harmonic Language Models," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Paris, France, 2018, 10–17.

[22] C. L. Krumhansl, *Cognitive foundations of musical pitch*, Vol. 17, Oxford University Press, 2001.

[23] K. Lee and M. Slaney, "Acoustic Chord Transcription and Key Extraction From Audio Using Key-Dependent HMMs Trained on Synthesized Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2), 2008, 291–301, ISSN: 1558-7924, DOI: 10.1109/TASL.2007.914399.

[24] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 2019.

[25] M. Mauch and S. Dixon, "Simultaneous Estimation of Chords and Musical Context From Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6), 2010, 1280–9, ISSN: 1558-7924, DOI: 10.1109/TASL.2009.2032947.

[26] B. Mcfee and J. P. Bello, "Structured Training for Large-Vocabulary Chord Recognition," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Suzhou, China, 2017, 188–94.

[27] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "An End-to-End Machine Learning System for Harmonic Analysis of Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 20(6), 2012, 1771–83.

[28] H. Papadopoulos and G. Peeters, "Joint Estimation of Chords and Downbeats From an Audio Signal," *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1), 2011, 138–52, DOI: 10.1109/TASL.2010.2045236.

[29] J. Pauwels and J.-P. Martens, "Combining Musicological Knowledge About Chords and Keys in a Simultaneous Chord and Local Key Estimation System," *Journal of New Music Research*, 43(3), 2014, 318–30.

[30] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 Years of Automatic Chord Recognition from Audio," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Delft, The Netherlands, 2019, 54–63.

[31] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 77(2), 1989, 257–86.

[32] H. Schreiber and M. Müller, "Musical Tempo and Key Estimation using Convolutional Neural Networks with Directional Filters," in *Proceedings of Sound & Music Computing Conference (SMC)*, Málaga, Spain, 2019, 47–54.

[33] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, "Audio Chord Recognition with a Hybrid Recurrent Neural Network," in *proceedings of the International Society for Music Information Retrieval (ISMIR)*, Malaga, Spain, 2015, 127–33.

[34] Y. Wu, T. Carsault, E. Nakamura, and K. Yoshii, "Semi-Supervised Neural Chord Estimation Based on a Variational Autoencoder With Latent Chord Labels and Features," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2020, 2956–66.

[35] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF Sequence Decoding Model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(2), 2019, 355–66, ISSN: 2329-9290, DOI: 10.1109/TASLP.2018.2879399.

[36] Y. Wu, T. Carsault, and K. Yoshii, "Automatic Chord Estimation Based on a Frame-wise Convolutional Recurrent Neural Network with Non-Aligned Annotations," in *proceedings of 27th European Signal Processing Conference (EUSIPCO)*, 2019.

[37] Y. Wu, E. N. Nakamura, and K. Yoshii, "A Variational Autoencoder for Joint Chord and Key Estimation from Audio Chromagrams," in *proceedings of Asia-Pacific Signal and Information Processing Association (APSIPA)2020*, 2019.