creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사 학위논문

음악적 요소에 대한 조건부 생성의 개선에 관한 연구: 화음과 표현을 중심으로

Improving Conditional Generation of Musical Components: Focusing on Chord and Expression

2023 년 2 월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

유 승 연

공학박사 학위논문

음악적 요소에 대한 조건부 생성의 개선에 관한 연구: 화음과 표현을 중심으로

Improving Conditional Generation of Musical Components: Focusing on Chord and Expression

2023 년 2 월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

유 승 연

음악적 요소에 대한 조건부 생성의 개선에 관한 연구: 화음과 표현을 중심으로

Improving Conditional Generation of Musical Components: Focusing on Chord and Expression

지도교수 이 교 구

이 논문을 공학박사 학위논문으로 제출함

2023 년 1 월

서울대학교 융합과학기술대학원

융합과학부 디지털정보융합전공

유 승 연

유승연의 공학박사 학위논문을 인준함

2023 년 1 월

위 원 장 곽 노 준 (인)

부위원장 이 교 구 (인)

위 원 이 원 종 (인)

위 원 남 주 한 (인)

위 원 이 경 면 (인)

Abstract

Conditional generation of musical components (CGMC) creates a part of music based on partial musical components such as melody or chord. CGMC is beneficial for discovering complex relationships among musical attributes. It can also assist non-experts who face difficulties in making music. However, recent studies for CGMC are still facing two challenges in terms of generation quality and model controllability. First, the structure of the generated music is not robust. Second, only limited ranges of musical factors and tasks have been examined as targets for flexible control of generation. In this thesis, we aim to mitigate these two challenges to improve the CGMC systems. For musical structure, we focus on intuitive modeling of musical hierarchy to help the model explicitly learn musically meaningful dependency. To this end, we utilize alignment paths between the raw music data and the musical units such as notes or chords. For musical creativity, we facilitate smooth control of novel musical attributes using latent representations. We attempt to achieve disentangled representations of the intended factors by regularizing them with data-driven inductive bias. This thesis verifies the proposed approaches particularly in two representative CGMC tasks, melody harmonization and expressive performance rendering. A variety of experimental results show the possibility of the proposed approaches to expand musical creativity under stable generation quality.

Keywords: Music generation, symbolic music, conditional music generation, melody harmonization, performance rendering, musical structure, musical hierarchy, controllability, latent representation, regularization

Student Number: 2015-31346

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Music is a representative content that entertains our daily lives and inherits human creativity. It can be created through an intelligent process by both human composers and instrument performers. Human composers utilize musical components to build structural and expressive ideas based on deterministic rules and personal inspiration [1, 2]. Furthermore, written music is delivered as sound by human performers who express their own understanding of music through behavioral skills [3]. Music generation encompasses these creative processes by composers and performers, as illustrated in Fig. 1.1. *Automatic music generation* is the attempt to mimic the music generation process of humans using computational methods [4, 5]. Recent studies have actively tackled music generation by mitigating the challenges in various applications in this field [6].

Applications of music generation can be divided into two categories by the target domain: symbolic music content and audio content. Symbolic music content primarily appears as a visual expression that demonstrates high-level mu-

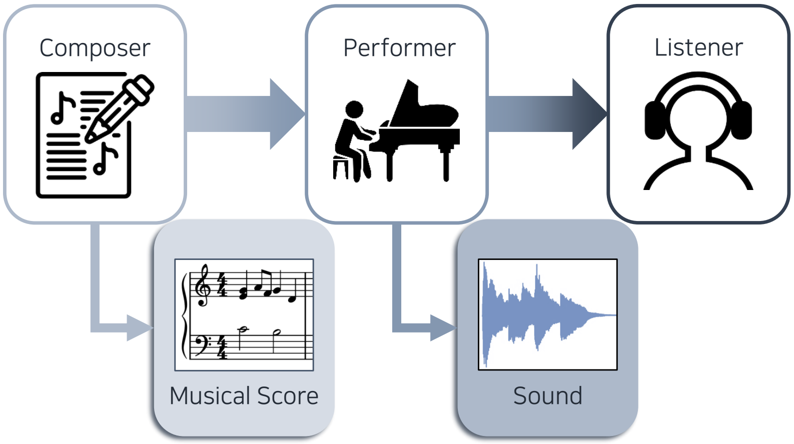


Fig. 1.1 Concept of music generation.

sical concepts such as pitch and rhythm. Examples include a musical score produced by a composer, such as those often seen in high school classes, or computational data in the form of a Music Instrument Digital Interface (MIDI). On the other hand, audio music content is an actual sound conveyed by a vibrating signal, and it is a complex manifold of low-level attributes that can be directly understood by the human ear [5]. Attempting to generate both symbolic music and audio in sequence is analogous to conducting the entire process of music generation from scratch, as illustrated in Fig. 1.2 [5, 7].

Among these two different domains, this thesis focuses on generating symbolic music. Generating the waveform of music may be more friendly to real-world use cases as it would not need any additional rendition process to convey the sound [8]. Nonetheless, a deep investigation of symbolic content is also significant for building an elaborate, domain-based system that can create coherent music. In particular, symbolic music content directly represents musical attributes that can help researchers constructively explore the organization of

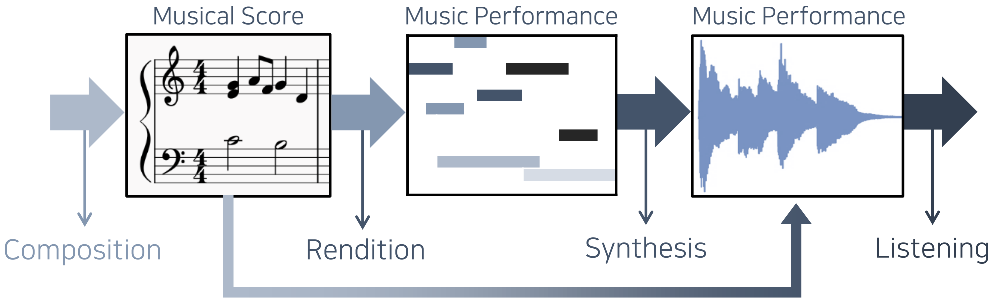


Fig. 1.2 Diagram of music generation process.

music that can be generally understood by humans [9].

The generation of symbolic music includes two stages: score generation and performance generation. According to Fig. 1.2, the first stage generates a musical score which provides basic information about the musical notes. As a music passage is composed of musical notes that are connected in both horizontal and vertical relationships, score generation needs to model complex relationships among structural attributes of the notes, such as pitch, rhythm, or chord. Some corresponding tasks include generating a monophonic melody [10, 11], percussive sequence [12], polyphonic music [13, 14], and multi-track music [15]. The second stage is generating an expressive music performance based on the score. The musical score should be conveyed to an actual sound to be heard by humans [7]. This conversion can be executed by a human performer who physically delivers the loudness and timing of each note in a musical piece. Hence, music performance generation aims to imitate this highly complex mechanism of a human performer that is connected to human perception and understanding of music [3]. Conventional studies have sought to model expressive dynamics [16] and expressive timing [17], and to generate a realistic performance [18], from given Western musical scores.

Each stage of symbolic music generation can be further divided according to whether the music is generated with constraints [6]. Music generation without any constraints, or from scratch, refers to an end-to-end production of low-level musical components that form an entire piece of music [5]. On the other hand, music generation with constraints requires the imposition of an additional condition to generate the music. This type of generation has been referred to as *conditional* music generation. Conditional music generation is useful in that it can satisfy a listener's needs, taking into account the nature of music that is consumed based on one's preference. For example, a listener may desire to listen to music in a bright mood that is suitable for a sunny afternoon. The condition would be the label of the corresponding mood, "bright", and the model is constrained to generate music that matches with the label. Fig. 1.3 illustrates the concept of conditional music generation and the usage of the corresponding system.

Conditional music generation finds or generates appropriate music under certain constraints. Constraints are given to the system as additional inputs that represent the meta-information or specific components of music. The constraints of meta-information, including emotion [19, 20], style [21, 22], or instrument [23, 24], fundamentally determine a taxonomy of music. On the other hand, the constraints of musical components, including melody [25, 26, 27, 28], or chord [29, 30, 31] provide the least boundary that the remaining part of the music can develop coherently. There are also other components that derive from other domains such as lyrics or image [32, 33]. In this thesis, we focus on the conditional generation of symbolic music given the musical components. This research topic

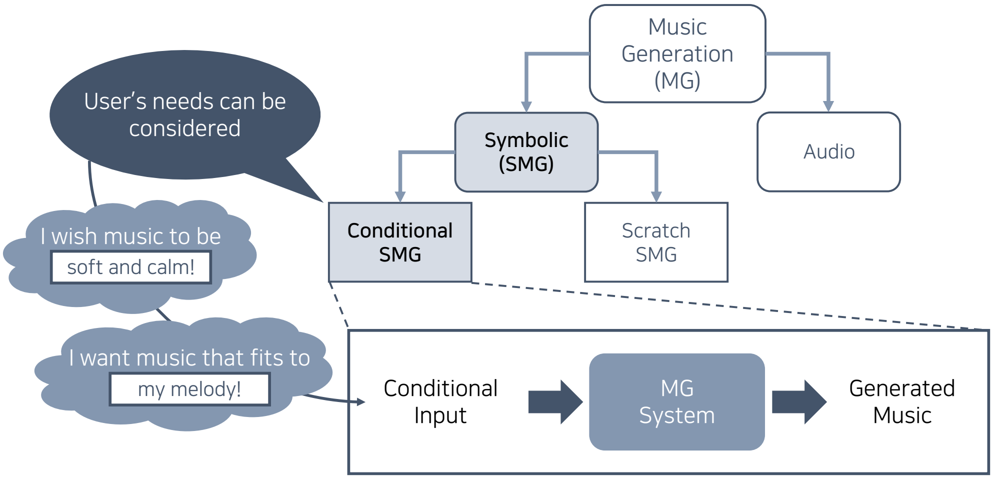


Fig. 1.3 Concept of conditional symbolic music generation (SMG).

aims to generate the remaining musical components given certain musical components as the explicit condition of the generative model. We abbreviate this topic as *CGMC*, representing "**c**onditional **g**eneration of **m**usical **c**omponents" for simplicity throughout this thesis. The following chapter is organized as follows. Chapter 1.1 demonstrates the necessity of investigating CGMC. Chapter 1.2 introduces definitions and keywords that are frequently used to represent musical components in a musical score and performance. Chapter 1.3 describes the main goal of the thesis with respect to the challenges in CGMC. Chapter 1.4 explains the methodologies that are utilized to achieve the main goal of the thesis. Finally, Chapter 1.5 summarizes the outline of the thesis.

1.1 Motivation

Conditional generation of musical components (CGMC) requires finding the remaining parts that musically match the given components of the target music.

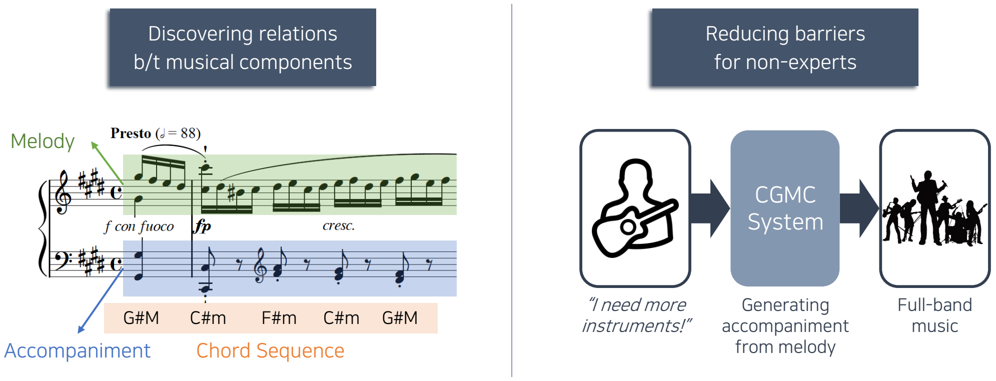


Fig. 1.4 Academic and commercial purposes for developing CGMC systems.

This framework may be less challenging than generating music from scratch, as the conditional inputs may reduce the uncertainty that the system must overcome. Nonetheless, CGMC has received active attention from a number of studies due to two reasons, as illustrated in Fig. 1.4.

Firstly, conditional generation of music focuses on discovering complex relationships among musical components [1]. Real-world music is composed of multiple components that are interrelated with each other to complete a pleasant sound. Therefore, it is necessary for the system to understand a complex manifold of these components to understand and reconstruct realistic music. In particular, the musical components are connected not only temporally but also vertically, weaving complex textures [34]. Hence, this framework can factorize such manifold in music by attempting to mathematically imitate the way that humans leverage each musical component. Thus, it has been often investigated as a downstream task of generating entire music from scratch [14, 35, 34].

Another reason is that conditional music generation given musical components can reduce barriers to creating music for users with less expertise [1]. Due

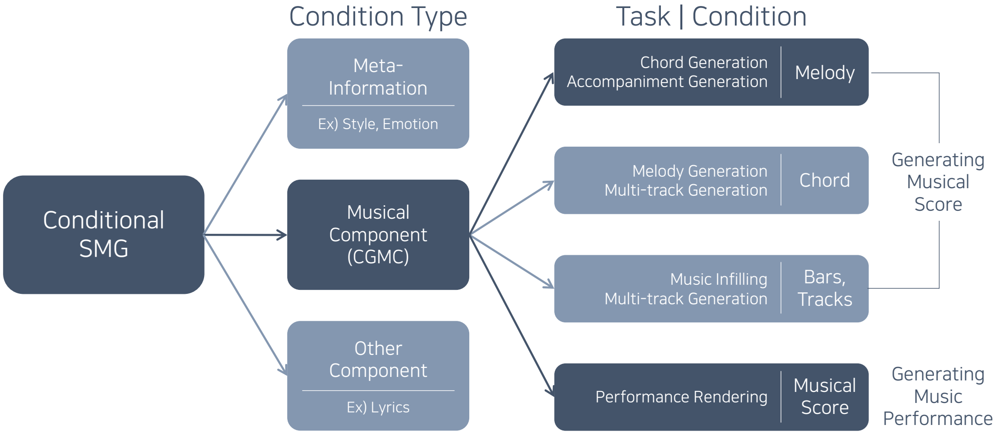


Fig. 1.5 The overall taxonomy of CGMC. Dark blue boxes are the particular area that this thesis focuses on, which are demonstrated in Section 1.4.3.

to the complex nature of music, a person who has not been trained may face difficulty at any level in making the whole music at once, either in the form of a musical score or performance. For example, some people can instantly create melody lines by humming themselves or can find famous common chords that are easy to access online. However, they may need help to complete their music by determining appropriate harmonies for the melody or filling in missing tracks [36]. The systems for conditional music generation can support users who desire to create music by their own tastes by providing the remaining parts of the music based on the elements given by the users.

CGMC has regarded various musical components derived from either a musical score or an expressive performance as the target outputs. Fig. 1.5 illustrates the overall taxonomy of the CGMC field that starts from the conditional SMG. For creating a musical score, a part of the score such as melody, chord, surrounding bars, or track of music have been given to the system. When a melody is given, the corresponding tasks are melody harmonization tasks [36, 26], or accompaniment generation tasks [37, 28]. Chord-conditional generation creates a melody [30], polyphonic music [38], or multi-track music [39] that follows the given chord progression. A music infilling task aims to fill in the missing bars of music given the surrounding bars [40]. Finally, a single track of music can be provided as the condition for multi-track generation [15].

In the case of rendering an expressive music performance, an entire musical score can be given to the system [41]. We use the term "rendering" distinguishing it from the term "generating" which can denote an end-to-end generation of a music with expressiveness from the scratch [7]. The musical score contains a structure of musical notes, a symbol that represents a sound, as well as various instructions for how to convey the corresponding music [2]. Hence, it is analogous to a text-to-speech (TTS) task where an actual sound should be derived from static symbols that appear as a text. The target parameters for rendering the music performance include expressive dynamics [16, 42, 43, 44], timing [45, 17], or the entire parameters to render the whole performance [46, 47, 48].

1.2 Definitions

Prior to describing the main goals of this thesis, we introduce definitions of musical concepts that this thesis mainly considers. The descriptions are presented according to whether the corresponding concepts are from a musical score or an expressive performance.

Concepts for Musical Score. The concepts that describe a single musical note are as follows: **Pitch** is a perceptual attribute that is derived by the periodicity, or fundamental frequency (F0), of sounds [49]. It is distinguished in the way that a sound is perceived high or low. Pitch can be represented by a combination of two attributes: pitch-class and octave. **Pitch-class** denotes 12 symbolized classes for the perceived height of a sound, quantized by equal temperament. The classes are "C", "C#", "D", "E", "F", "F#", "G", "G#", "A", "A#", and "B". While the pitch-class represents the harmonic color of the sound, **octave** denotes the absolute level of height that increases or decreases by every 12 semi-tone intervals. **Onset** and **offset** are the time when a note starts and ends, respectively. **Duration** is the length of a note's sustain, which is also the distance from the onset to the offset of the note. **Inter-onset interval (IOI)** is the time interval between two note onsets. An IOI of a target note can be computed based on either the previous or adjacent note of the target note. Higher-level concepts than a single note are as follows: **Melody** is a monophonic sequence of musical notes that a listener may perceive and remember as the primary contour or essence of the entire music [50]. **Chord** is a group of notes that are heard simultaneously [51]. Within a chord, the notes inherit vertical relationships among each other. Depending on the literature, a chord can represent roughly two concepts: the notes that are hit simultaneously by one or more voices or instruments [52], or a harmony that musically fits the melody and forms a background of the music [1].

Concepts for Expressive Performance. The common concepts for music expression are as follows: **Dynamics** is the loudness of a note, which determines how loud or soft the note is perceived [53]. Thus, a group of notes shapes the overall loudness of the music. The overall loudness can also vary over time: dynamics getting louder over time is called *crescendo*, and vice versa is called *diminuendo.* In contrast, dynamics can vary for a single note to emphasize the note. This is called an *accent.* **Articulation** denotes how short or long the note is played compared to the distance between two successive notes [54]. The note is considered to be in stronger *staccato* if the silence gets longer between the two notes. **Tempo** is how fast a note is performed. The overall tempo is determined by the density of the temporal grid on which the notes are played. Tempo can also change locally for individual notes when a performer intensifies the expressivity [54, 7]. This technique is called *rubato.*

1.3 Tasks of Interest

In this section, we introduce the tasks of interest that we tackle throughout this thesis. Concretely, we focus on improving CGMC systems in a way their products can be consumed and utilized as much as human-composed music. Recent studies for CGMC have achieved promising results in generating relevant musical components with respect to the conditional component. Nonetheless, these studies have left room for creating realistic, innovative music beyond mimicking the existing data [9]. To this end, we tackle two challenges with respect to enhancing the inner quality and creativity of the generated music.

1.3.1 Generation Quality

One challenge in CGMC is to increase quality of the generated music. Improving generation quality involves choosing appropriate model architecture and data representations that can help the computational model effectively learn the structure of music. Conventionally, music has been considered to be sequential data along with language [55]. Hence, a number of music generation studies have leveraged language models or other sequential models to achieve good performance in generating both short-term and long-term music [10, 7, 31].

However, music is in fact distinguishable from natural language in that it inherits vertical relations forming multi-dimensional textures [34]. Therefore, all musical notes cannot be simply serialized into a 1-dimensional sequence, where the entries should be grouped differently according to whether they are connected temporally or harmonically. Moreover, most data representations have not been intuitive to reflect meaningful musical concepts. Some common representations are pianoroll-based and event-based, where each entry describes a quantized time or a single event for playing a note [5]. In other words, multiple entries should be associated to completely represent a musical note. Such representations can be hardly intuitive when sequentially modeling between adjacent musical notes.

Furthermore, most CGMC studies have not deeply investigated an explicit method to capture the structuredness of music. Some studies have proposed novel methods to capture regular metrical structures or hierarchies in musical attributes [56, 57]. Nonetheless, these methods are built based on sequential models that fall short of reproducing the repetitive patterns, such as a hidden Markov model or a recurrent neural network [56, 58]. Some CGMC systems have employed Transformer-architecture [59], and they may have improved structuredness of the generated music [35]. However, they have still used the less-intuitive data representations that can weaken the model strength to encode proper musical structure.

1.3.2 Controllability

Another challenge in CGMC is to improve the controllability of generated music. Controllability of music can deeply satisfy listeners or music makers who desire to create music that reflects their specific tastes. The output can be controlled through conditional inputs, ranging from high-level attributes such as genre to low-level attributes such as a note sequence itself [9]. These conditional inputs determine the boundaries of the output distribution, therefore the representation of the output is strongly entangled with the condition. If the conditional input is the only method given to the model to control the output, the output may not be modified further to maintain the bond with the condition.

A user may wish to control various attributes of a piece of music, independent of the specified conditions. For example, a user may desire to increase the rhythm density of a created melody, while maintaining its chord progression, as specified by the condition [60]. This level of control is achievable when the system is able to independently manipulate particular attributes of the data without disrupting other attributes that are connected to the condition [61]. In a deep learning framework, this independent control can be implemented with *semantic* representations learned from the training data and the conditional inputs. Representations can be semantic when certain axes are meaningful with respect to high-level attributes that are understandable to humans [62]. Achieving good representations is often analogous to learning *disentangled* representations, where each representation is sensitive to a specific factor of the target data, while not being sensitive to others [63, 64]. In various domains such as images or speech, recent studies have attempted to build generative models that can learn disentangled representations from the hidden attributes of the data and meaningfully control the outputs [62, 65, 61].

Music generation studies have concurred with this trend and proposed generative models that can learn meaningful representations of musical attributes [66, 15, 31]. These attempts have facilitated the flexible modification of certain attributes, such as rhythm or pitch contour, leading to an enhancement of musical creativity [67, 60]. Certain CGMC systems have also successfully achieved controllability through meaningful latent representations [29, 57]. However, these attempts are still in their infancy [68], and only a limited range of musical attributes and tasks have been explored as controllable factors and target tasks, respectively. Some of the most common attributes that have been studied as controllable factors include rhythmic pattern or density [67, 31], note density [69, 60], chord and texture [29, 70]. Furthermore, controllable generation using the latent representations has been applied to various tasks including melody generation [67, 71], style transfer [29, 72, 69], music inpainting [40, 73], and polyphonic music generation [60].

1.4 Approaches

In this section, we introduce concrete approaches that we use to tackle the task of interest. The first subsection demonstrates how we improve the structuredness of the generated music using intuitive perspectives in terms of musical concepts. The second subsection describes the existing approaches that we choose to achieve semantic representations for the target attributes for enhancing the controllability of CGMC systems. Lastly, the third subsection introduces two target tasks that we deal with throughout this thesis.

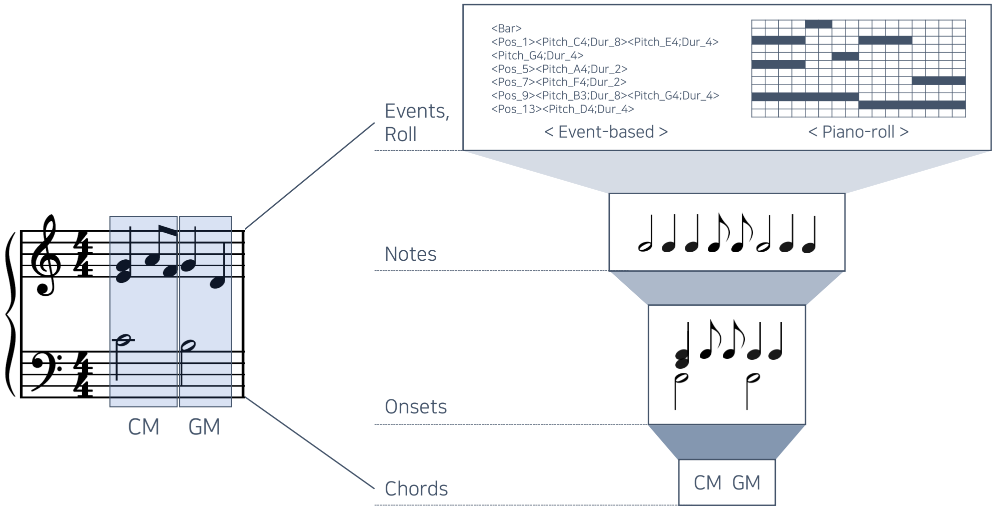


Fig. 1.6 An example of musical hierarchy derived from a music excerpt.

1.4.1 Modeling Musical Hierarchy

Musical structure is deeply related to the hierarchy of music, as human perceives the structure of the musical passage by grouping it into higher-level units such as motives, phrases, and sections [74]. The musical hierarchy includes elaborate relationships among multiple semantic units of sound that represent common musical concepts, as illustrated in Fig. 1.6. The smallest unit in musical score may be a note, which is represented as a black oval drawn on the 5-line staff. On the other hand, the highest unit can be a part or phrase representing a passage of sufficient length that is analogous to a sentence or paragraph in language.

In contrast, the music generation field considers the smaller unit than the note for representing the raw music data: a temporal frame or an event [5]. As described in Section 1.3.1, musical data has been often parsed as a sequence of event-based tokens or a 2-dimensional piano roll. In particular, event-based tokens represent musical events such as playing a note, stopping a note, temporal

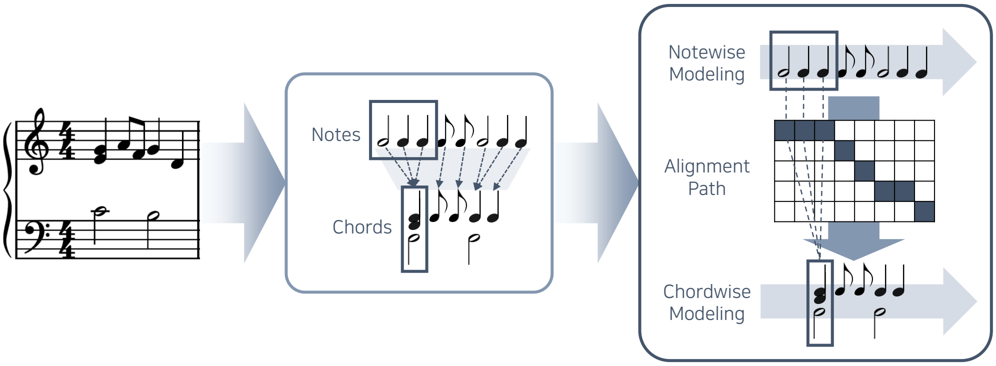


Fig. 1.7 Overview of the approach using an alignment path to capture musical hierarchy.

shift, dynamics, etc [7]. Furthermore, recent studies have adopted variants of this representation that utilizes additional information such as position, tempo, instrument, and chord [39, 28, 75]. Meanwhile, the piano-roll is a matrix with two axes which represent the MIDI pitch number and time, and its non-zero entries represent the presence of note in the corresponding pitch [76, 15, 77]. The aforementioned representations are not intuitive for the generative model to learn dependency in musically meaningful units.

Inspired by recent studies that have addressed this issue [69, 78, 18], we propose to learn the hierarchical information by the effective encoding of the data representation. To this end, we use an alignment path between the raw representation and a sequence of meaningful units. If we know how the data representation is mapped to higher-level units such as notes or chords, we can make a binary matrix that reflects a hard alignment path. This matrix can be directly multiplied by the embeddings of the input data to produce an output where each timestep represents the target high-level unit: a vector in each timestep reflects accumulated information of the corresponding range of the input data. Then, this output can be divided by the number of aggregated inputs to reflect the average information for each target unit. The entire process is illustrated in Fig. 1.7.

Using this approach, the model can directly learn temporal dependency between notes, chords, and other meaningful units represented by the embeddings. This approach is expected to improve generation quality through capturing the explicit structure configured by the musical units and the clear relationship among the multi-level units.

1.4.2 Regularizing Latent Representations

To expand the controllability of the CGMC system, we use methods that regularize latent representations to align them with the desired factors. In particular, these methods are based on unsupervised or self-supervised learning frameworks that utilize the features derived from the data itself, instead of human-labored annotations within the musical data. These features can be directly used for supervision as the target labels, or they can implicitly provide the inductive bias that gives some information bottleneck to the model. These kinds of regularization frameworks have been often employed in various domains together with recent stochastic models, such as variational autoencoder, that map the hidden attributes of the data to the continuous latent space [79, 80].

As numerous attributes of music and human behaviors are entangled with each other, some kinds of inductive bias should be needed to achieve the intended representation from real-world music [64]. The inductive bias can be often given as annotations or labels for supervision. However, common annota-

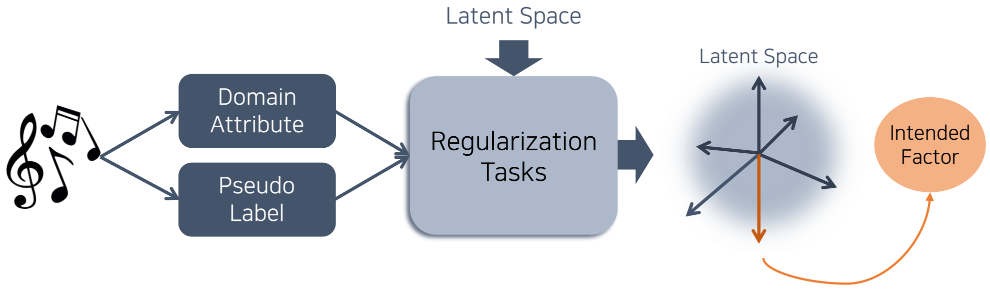


Fig. 1.8 Overview of the approach regularizing a latent representation to control musical components.

tions provided by human labor have been limited to meta information of music such as genre or artist. Especially, it would be hard to find the existing annotations that precisely represent the behavior of the target attributes when we want the system to control certain attributes of music in a non-static way. Therefore, it can be more effective to extract the desired features directly from the music data or force the latent representation with the inductive bias apparently derived from the data. Recent music generation studies have chosen this kind of regularization framework that utilizes the domain knowledge to disentangle certain factors related to musical characteristics. [67].

Achieving semantic representations allows the model to get controllability where the target attributes can be flexibly modified in the continuous space [66]. That is, the attributes can be either interpolated or modulated by time using the latent representation as if a user "slides a knob" of the DJ mixer [60]. To enable this continuous control of the attribute, we particularly refer to the methods by Pati *et al.* [68, 81, 82]. Pati et al. have proposed algorithms for disentangling latent dimensions by restricting them to be aligned with certain musical attributes. Using this approach, the latent dimensions can function as knobs for disentangled attributes. Modifying the value in each dimension can result in a change of the aligned attribute toward the expected direction. In addition to this approach, we employ some auxiliary tasks to promote the disentanglement of latent representations For example, we conduct prediction tasks where the target latent representations are directly used with the self-derived *pseudo* labels [80]. The overall procedure of regularizing a latent representation to control a desired musical factor is demonstrated in Fig. 1.8.

1.4.3 Target Tasks

We choose two target tasks to attempt the aforementioned approaches: melody harmonization and performance rendering. These two tasks are selected for two reasons. First, controllable generation has not been widely explored in these tasks. Second, they have been the representative downstream tasks with respect to the musical score and expressive performance, respectively, in the music generation field. By studying how to improve performances on these downstream tasks, we expect to build basic knowledge on developing music generation systems that can create real music.

Melody Harmonization. One of the main goals of this task is to find a coherent chord sequence that harmonically matches to the given melody. The aforementioned challenges have not been deeply tackled especially in this task. Chords are the high-level attributes that significantly determines the semantics in a musical passage. Hence, the melody harmonization task has been one of the common downstream tasks in music generation. This leads to an importance of tackling this task: improving the chord structure at downstream task can be crucial for enhancing the generation quality of music generation systems that particularly need explicit chord information. Moreover, it can also promote musical creativity by modifying the harmonic texture of music.

Performance Rendering. This task mainly aims to generate an relevant set of expressive attributes, such as dynamics, articulation, and tempo, from a written musical score. Exploring this task is important as it focuses on discovering relationships between a static score and human behavior of delivering music [41]. Particularly in the case of music generation, it has been a challenging issue to evaluate most of the systems as their new music should be delivered with the actual sound for evaluation [7]. Hence, this task is an essential downstream task for developing music generation systems. We also target all expressive attributes at once for the effective rendering of expressive performance, rather than generating only a part of the attributes. This can facilitate controllability in all attributes, different from the previous studies.

1.5 Outline of the Thesis

This section summarizes the outline of this thesis as the following studies. Fig. 1.9 also demonstrates how these studies correspond to the tasks of interest aforementioned in Section 1.3.

Chapter 2 provides backgrounds and literature for the task of interest and the core concepts considered throughout this thesis. In the first section, we introduce previous studies for the target tasks, which are melody harmonization and expressive performance rendering. The next section describes the attempts to solve the first challenge which is improving the musical structure. The third section introduces the concept and types of disentanglement learning, which

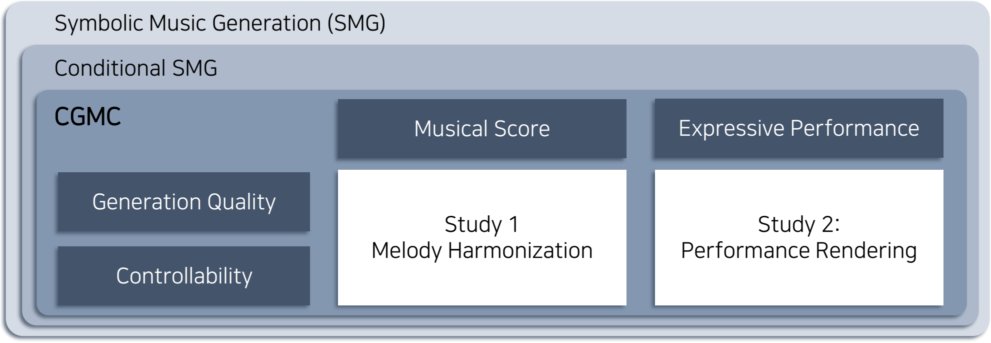


Fig. 1.9 Overview of this thesis.

is related to solving the second challenge, and the corresponding studies. Finally, we describe the studies for controllable music generation that uses the framework of regularizing latent representations.

Chapter 3 discusses methods to improve structuredness and controllability of melody harmonization system. Recent deep learning approaches for melody harmonization have achieved remarkable performance by overcoming the uneven chord distributions of music data. However, most of these approaches have not attempted to capture an original melodic structure and generate structured chord sequences with appropriate rhythms. Hence, we use a Transformer-based architecture that directly maps lower-level melody notes into a semantic higher-level chord sequence. In particular, we encode the binary piano roll of a melody into a note-based representation. Furthermore, we address the flexible generation of various chords with Transformer expanded with a VAE framework. We propose three Transformer-based melody harmonization models: 1) the standard Transformer-based model for the neural translation of a melody to chords (STHarm), 2) the variational Transformer-based model for learning the global representation of complete music (VTHarm), and 3) the variational Transformer-based model that is regularized for the controllable generation of chords (rVTHarm). Experimental results demonstrate that the proposed models generate more structured, diverse chord sequences than LSTM-based models.

Chapter 4 presents a novel system for rendering a symbolic piano performance with flexible musical expression. It is necessary to actively control musical expression for creating a new music performance that conveys various emotions or nuances. However, previous approaches were limited to following the composer's guidelines of musical expression or dealing with only a part of the musical attributes. We aim to disentangle the entire musical expression and structural attribute of piano performance using a conditional VAE framework. It stochastically generates expressive parameters from latent representations and given note structures. In addition, we employ self-supervised approaches that force the latent variables to represent target attributes. Finally, we leverage a two-step encoder and decoder that learn hierarchical dependency to enhance the naturalness of the output. Experimental results show that our system can stably generate performance parameters relevant to the given musical scores, learn disentangled representations, and control musical attributes independently of each other.

Chapter 2

Background

In this chapter, we introduce conventional attempts related to the target tasks and frameworks that we focus on throughout this thesis. Concretely, we explore previous efforts that aimed to generate realistic chord sequences and expressive performance. Moreover, we revisit recent studies in music generation tasks that have addressed the two challenges: improving generation quality and controllability of the generative models. To this end, we compose this chapter by four categories: music generation tasks; approaches to enhance structure in music generation; disentanglement learning; and controllable music generation. As we aim to tackle two challenges with respect to generation quality and controllability, we demonstrate backgrounds according to these two topics.

This chapter is organized as follows. The first section introduces the existing approaches for the two target tasks. The second section provides concrete attempts to enhance structure in music generation. In the third section, we explain common definition of disentanglement learning and introduce studies for unsupervised, supervised, and self-supervised learning. Lastly, we present studies for controllable music generation that mostly use self-supervised framework.

2.1 Music Generation Tasks

In this chapter, we demonstrate previous approaches for generating the two musical components: chord label and expressive parameters. The corresponding tasks have been called as melody harmonization and expressive performance rendering, respectively. We explore these conventional studies to revisit the limitations with respect to the two challenges that we are tackling in this thesis. First subsection introduces the conventional studies of melody harmonization that employed various methods from the rule-based to the recent deep learning methods. In the second subsection, we also introduce a brief history of the approaches for expressive performance rendering.

2.1.1 Melody Harmonization

Melody harmonization generally aims to find musically plausible chord sequence given a sequence of melody. A target chord sequence can be represented as chord labels or polyphonic voices that form a Bach's Chorale with four polyphonic voices [1]. In the latter case, a given melody is mostly analogous to the topmost voice of the Chorale, or Soprano. Conventional studies have tackled both tasks with various approaches, starting with the rules from Western classical music related to harmonic grammar.

Rule-based studies aim to simulate structural chord progressions carefully using linguistic techniques and heavy domain knowledge [83, 84, 85, 86]. Generic algorithms (GAs) were early probabilistic solutions that were combined with rule-based constraints [87, 88]. Machine learning approaches such as hidden Markov models (HMMs) demonstrate the use of probabilistic modeling to assess temporal dependency in music [36]. However, due to the inability of a standard HMM to capture elaborate harmonic functions, the HMM-based model was improved with domain knowledge [89] or tree-structured Markov models based on probabilistic context-free grammar [56, 90].

Lim *et al.* [26] utilized a stacked bidirectional long short-term memory (BLSTM) model to predict a chord for each bar of a given melody that was aggregated into a pitch-class histogram. This LSTM-based approach successfully improved model robustness for the skewed distribution of commonly used chords. Recently, Yeh *et al.* [25] revisited a sufficient number of conventional methods and consequently proposed MTHarmonizer, a deep multitask model that predicts chords with correct phrasings by directly supervising harmonic functions. Canonical metrics for assessing the coherence and diversity of the created chord sequence were also proposed. Sun *et al.* [91] used the orderless neural autoregressive distribution estimation (NADE) and the blocked Gibbs sampling method to approximate the complex joint probability among chords given a melody. They provided the model with a masked chord sequence so that the model could predict masked entries and leveraged class weights to efficiently balance the uneven distribution of chords.

These LSTM-based models shared the same data representation and model architecture. In particular, the models by Yeh *et al.* and Sun *et al.*, which improved the musical grammar or the diversity of chord types, were extensions of the model by Lim *et al.*. However, we assume that the LSTM-based approach is limited to modeling a serialized chord sequence without capturing the realistic pattern of chords from an unintuitive encoding of the melody structure. In this thesis, we investigate the intrapatterns and interrelationship of the melody and chords.

2.1.2 Expressive Performance Rendering

The conventional studies on computational modeling of music performance have aimed to connect the physical parameters of the performance, such as loudness or timing, to the symbolic attributes of the music scores. One of the early approaches was to construct the algorithmic models based on some theories in music performance and to validate the models by listening to the model-generated performances [92]. The KTH rule system following this approach intensively confirmed the various rules that reflect the previous findings in the music cognition studies as well as the musical domain knowledge [92, 93].

With the advance of machine learning techniques, the rules of music performance were computationally extracted from the actual data to simulate the human performance. YQX system used a simple Bayesian model to find the relationships between the musical score context and target performance parameters related to timing, dynamics and articulations [46, 54]. An ensemble learning method was also used to discover the general principles in expressive music performance [94]. Gaussian process (GP) regression simplified the parametric rules to predict the expressive parameters [95]. A Markov model with switching Kalman filter was applied to parse the musical interpretation with the discrete and continuous parameters that represent expressive timing [96]. Linear and non-linear basis models were deeply investigated as a new paradigm for predicting the expressive dynamics and timing [16, 42]. Conditional random fields (CRFs) were used for stochastic generation of expressive piano performance [97]. *Shunji* was a feed-forward system based on the case-based reasoning paradigm, which stored the performance segments and found the most similar segment to the new score input [98]. Furthermore, deep learning methods such as a recurrent neural network (RNN) and a graph neural network (GNN) encouraged the generative models to intuitively encode a large number of performance sequences and to create more realistic piano performances [99, 100].

More recently, deep probabilistic models such as variational autoencoder (VAE) have facilitated the stochastic generation of the performance parameters. In performance generation, some studies used conditional VAE (CVAE) that adapted the score attributes as the condition. Maezawa *et al.* initially attempted to use CVAE to render the piano performance from aligned score data in note-level [48]. Jeong *et al.* further showed a breakthrough achievement in generating the professional piano performance using CVAE which was improved with hierarchical recurrent architectures [18]. Most of these recent approaches have focused on reproducing realistic expressive performances that correspond well to the given score. The given musical score has provided rich information intended by the composers on how the score should be musically expressed. In other words, there has been less room for any potential listeners to control performance attributes flexibly beyond the existing classical scores.

2.2 Structure-enhanced Music Generation

In this section, we present recent attempts to improve structure in the generated music. We introduce them into two streams in terms of the type of the framework. The first subsection explains various approaches that modify or add model architectures within the existing frameworks such as Markov chain. The second subsection introduces recent studies using the current powerful framework, Transformer [59], which is empowered with attention mechanism. This trend has been initiated with the common supposition that music is a sequential data analogous to language.

2.2.1 Hierarchical Music Generation

Although it has been challenging to embed concrete structure in the generated music, a number of studies attempted to reproduce realistic music considering its temporal nature. Conventional studies have often utilized sequential models, such as Markov models or recurrent neural network, to learn musical progression in the way to learn the linguistic grammar. More recently, Transformer has become one of the most common framework for various music generation tasks, as it has shown powerful generation performance in the natural language processing (NLP) field [59]. While these frameworks could be effective on modeling temporal dependency, other approaches have also been employed to effectively learn musical hierarchy.

Some studies in melody harmonization attempted to improve chord structures by using auxiliary algorithms. Tshshima *et al.* exploited probabilistic context-free grammar (PCFG) to explicitly train hierarchical structure of functional harmony, while Markov models learned harmonic rhythms and generated the corresponding melody sequence [56, 90]. In melody generation task, Wu *et al.* proposed hierarchical RNN composed of three subnetworks modeling bar, beat, and note-level structure using long short-term memory (LSTM) modules [101]. Robert *et al.* also proposed MusicVAE where a Conductor module generates bar-level representation prior to the note-level decoder. This model was further developed for multi-track generation by Simon *et al.*, expanding the architecture to encode and decode track-level representations [102]. MuseGAN by Dong *et al.* enhanced the original generative adversarial network (GAN) by adding a module to encode bar-level representation to create semantic multi-track music [15].

Recent Transformer-based models also improved their generation quality by using VAE that learns bar-level representations [78, 69]. In performance rendering task, Jeong *et al.* proposed to learn measure-level representations from the serialized data of polyphonic piano performances using hierarchical attention network (HAN) composed of LSTM modules [18]. This enabled the model to effectively encode and decode musical hierarchy of long-term polyphonic piano performances, also saving the computational cost.

2.2.2 Transformer-based Music Generation

Transformer architecture has been widely exploited for various music generation tasks, as it has shown powerful performance in capturing and generating patterns within a sequential data [59]. In particular, multi-head self-attention mechanism in Transformer has intensively helped the model grasp structures of multiple levels in unsupervised way. Music Transformer, introduced by Huang *et* *al.* [14], was one of the most successful models for generation of long-term symbolic music. It was based on a novel event-based representation that effectively encoded polyphonic real-world music with expression [7].

LakhNES used the extended Transformer architecture, Transformer-XL, to generate plausible multi-instrumental game sound chips [103]. Pop Music Transformer also used Transformer-XL and a novel data representation called "revamped MIDI-derived events (REMI)" was proposed to consider the metrical structure for generating polyphonic pop music [39]. Jazz Transformer adapted REMI to jazz music to create long-term coherent jazz lead sheets [104]. More recently, chord conditioned melody transformer (CMT) leveraged Transformer decoders to generate a *grid-based* melody given a chord progression [30]. This work attempted to create a melody with proper rhythms that were well aligned with the given chords. This work was similar to the current interest of our study.

Furthermore, Choi *et al.* [35] proposed a Transformer-based autoencoder that achieved global representation for the musical contexts of polyphonic piano performance data. Jiang *et al.* [78] introduced a hierarchical Transformer VAE to learn context-sensitive melody representation with self-attention blocks, enabling the model to control the melodic and rhythmic contexts.

2.3 Disentanglement Learning

The generative models in the various domains have explored the way to find good representations to increase robustness and task performance of the model [63, 64]. Most of these approaches have aimed to make a *disentangled* representation, in which the representation should be sensitive to variation of a certain factor while invariant to other factors of the observed data [63]. It has been widely accepted by studies for generative models that achieving disentangled representation of a certain factor is analogous to getting controllability of that factor [105, 106, 82]. In the rest of this section, we present a concept of disentanglement learning and the corresponding approaches in three categories: unsupervised, supervised, and self-supervised approaches.

2.3.1 Unsupervised Approaches

The unsupervised approaches have used several methods to discover meaningful axes that represent the data distribution: enhancing the information bottleneck of the latent representation [107, 62], explicitly encouraging the independence among the latent variables [108, 109], or increasing the mutual information between the representation and data [110]. In particular, the unsupervised approaches attempted to factorize the observed sequential data into the time-invariant and time-variant factors. Text-to-speech generation models separated the acoustic attributes into linguistic and timbre-related factors using their architecture or distinctive objectives [65, 105, 111]. Video data was often decomposed into the content and motion factors by model architectures that differ by temporal dependencies of the factors [112, 61].

2.3.2 Supervised Approaches

The supervised methods have utilized the static labels to effectively constrain the latent representations. This is for capturing the target attributes that are hidden in the complex manifold of the data [64, 113]. A number of studies in the visual domain applied the supervised approaches that use the existing annotations [113, 114, 115] or the other inductive bias in data [116, 117, 118]. For text generation, Hu *et al.* enabled the semi-supervised training with partial annotations for sentence sentiment to disentangle the observed text into the sentiment and the remaining factors of the sentence data [119]. Some text-to-speech (TTS) generation tasks applied full or partial supervision to disentangle the global attributes of speech, such as age or emotion [120, 121, 122, 123].

2.3.3 Self-supervised Approaches

Limitations of the supervised approaches have led the recent studies for generative models to investigate the self-supervised learning framework. The supervised approaches have revealed poor robustness of the model from the noisy labels and the huge cost in human power [124]. In general, self-supervised learning methods have aimed to solve various pretext tasks such as image colorization [125], Jigsaw puzzles [126], and classification using self-generated labels [127, 128]. These self-derived labels, or *pseudo* labels, can be imposed by the domain knowledge related to the known factors of the data. They have assisted the model to discover the concrete boundaries of the data clusters and increase robustness in representation learning [124]. An example of the pseudo labels in dialogue generation can be the supervisory signals of topic or speaker which were extracted by the trainable modules for representation learning [79]. Some video generation studies decoupled motion factors from data by using the extracted keypoints [129] or the additional video data with the target motion [130]. S3VAE attempted to control both dynamic and static factors of sequential data by using the triplet loss and self-derived labels [80]. The multi-modal systems also applied the self-supervised framework to disentangle representations for video and audio. Nagrani *et al.* decomposed the speech data into speaker identity and linguistic content by using inter- and intra-track constraints from the speaking face tracks to promote speaker identification [131]. Rouditchenko *et al.* enabled audio-visual co-segmentation by predicting audio of the selected video from the mixed audio of two video streams [132].

2.4 Controllable Music Generation

In this section, we introduce recent approaches in music generation that have attempted to control musical attributes with disentangled representations. We first present the previous studies that aimed to flexibly generate components in a musical score. Those studies mostly have focused on disentangling latent representations that correspond to common musical attributes. In the next subsection, we explain several recent studies for rendering expressive performances. They also aimed to factorize real-world piano performance data into meaningful attributes.

2.4.1 Score Generation

In symbolic music generation, an increasing number of the systems have employed the discriminative learning frameworks that use self-supervision and domain knowledge. Pati and Lerch suggested a method for disentanglement learning that forces a latent variable to be monotonically related to one musical attribute [68]. EC^2-VAE decoupled rhythm and pitch attributes of a melody by achieving the rhythm representation through an intermediate supervision [31].

ExtRes applied the domain-based algorithms to extract musical features from data for learning the structured latent code [67]. MeasureVAE attempted to solve a "musical score inpainting" problem for melody generation by employing multiple VAEs learning the latent representations of the surrounding measures [40]. SketchNet also aimed to generate a missing measure by decoupling the latent representation of each surrounding measure into the factors of the pitch and rhythm [73]. FaderNet attempted to achieve the latent representations that bridge between music and the high-level attribute, arousal level, by learning the low-level attributes from the data-driven annotations [60]. Wang *et al.* used the hierarchical structure of music to disentangle chord and texture attributes from the polyphonic music [29]. Lastly, PianoTree VAE aimed to achieve the latent representation of the tree-structured musical syntax where the simultaneous notes were considered as a higher-level unit [38].

2.4.2 Performance Rendering

The recent studies in music performance generation have followed a similar trend. Maezawa *et al.* exploited the conditional variational recurrent neural network (CVRNN) framework to decouple a performer's interpretation from the corresponding musical score [57]. This study focused on the fact that the performer's interpretation could be affected by the piece-specific factors of the given music [133, 134, 135, 136]. The conditional priors that depended on the previous state of the RNN, were independent of the score attributes. The study validated that this assumption for the prior could support the model to capture the abstract representation of a performer's unique interpretation. Tan *et al.*, on the other hand, aimed to disentangle temporal dynamics and articulation of the performance data using the conditional priors and the intermediate predictions of the target factors [137]. The conditional priors for the two factors were derived from the trainable lookup tables that corresponded to the factors [105]. They also used the pseudo label sequences representing the onset-wise dynamics and articulation. The onset-roll, the matrix where the entries for the onset time were 1 and the remaining entries were 0, was provided as the condition of the system by the paired performance MIDI data.

2.5 Summary

In this chapter, we have provided an review of literature backgrounds that are closely related to our following studies. First of all, we have reviewed previous approaches that directly tackled melody harmonization and expressive performance rendering, which are the main tasks in our studies. Then, we have introduced the methods that recent music generation studies have exploited to enhance the musical structure of their products. We also have explained disentanglement learning that can expand the controllability of the generative model. We have presented the unsupervised, supervised and self-supervised approaches to learn disentangled representations. Finally, we have arranged the recent approaches that use stochastic models to disentangle and control the musical attributes of the generated music.

In the following chapters, we will discuss two main studies in detail. In Chapter 3, we describe three systems for melody harmonization that are improved on structuredness, diversity, and flexibility of the generated chords. In Chapter 4, we introduce a system for rendering expressive piano performances where novel factors of piano performance are hierarchically modeled and flexibly controlled by multiple latent representations. These two studies have a common goal of solving the two challenges of improving generation quality and controllability.

Chapter 3

Translating Melody to Chord: Structured and Flexible Harmonization of Melody with Transformer

3.1 Introduction

Automatic melody harmonization, which finds a coherent chord sequence that fits the given notes in a melody, is an essential topic in music generation. This task, which imitates the harmonizing process, is important for understanding human composition [1]. It is also practical for commercial use since it can reduce barriers to creating music without expertise [138, 36].

A melody harmonization task requires capturing the long-term dependencies in music since a constrained sets of chord progressions can consistently interact with a given melody [84]. This has motivated the use of linguistic techniques such as context-free grammar [83], genetic algorithms [88], or hidden

Markov models (HMMs) [36, 89, 56]. Recently, deep learning approaches with bidirectional long short-term memory (BLSTM) showed robust performance by effective nonlinear sequential modeling of bar- or half-bar-based melody and chords [26, 25, 91]. Moreover, these studies successfully overcame the uneven chord distributions that are in common musical data.

Nevertheless, these LSTM-based studies had limitations in generating concrete chord structures. First, the models were unable to encode an original melodic structure despite their sequential architectures [84]. The notes in a melody were aggregated within a chord duration into a pitch-class histogram before being fed to the model. Second, the models did not explicitly consider capturing the patterns of chord progressions. Chord labels correspond to the constant time grids (e.g., a bar or half-bar). Sequential modeling of grid-based chord labels is likely to result in ambiguous patterns or hierarchies of the generated outputs [56].

Hence, we attempt to utilize a recent language model, Transformer, for structured melody harmonization. Transformer directly encodes inter- and intra-structures between two sequential data in dynamic length [59]. Thus, with Transformer, we can approach melody harmonization as the *translation* between two different languages, melody notes and chord labels, which share a semantic musical context.

However, conventional Transformer-based studies encoded music as a series of musical events [14]. Using event-based representations differs from how humans perceive a rendered or score-written melody for harmonization [141]. Instead, a grid-based melody representation can be more intuitive for modeling melodic patterns synchronized with chord labels [84, 39, 30]. In our work, we

Model	Lim *et al.*[26]	Yeh *et al.*[25]	Sun *et al.*[91]	**STHarm**	**VTHarm**	**rVTHarm**
Backbone	LSTM	LSTM	LSTM	**Transformer**		
Chord Unit	one bar	half-bar	half-bar	half-bar		
Dataset	Wikifonia.org[26]	HLSD[139]	HLSD	HLSD & **CMD[140]**		
Key Signature	Normalized	Normalized	Normalized	Normalized & **Not Normalized**		
Chord Diversity	Yes	Yes	Yes	No	Yes	Yes
Structuredness	No	Yes	No	Yes	Yes	Yes
Stochasticity	No	No	No	No	**Yes**	**Yes**
Controllability	No	No	No	No	No	**Yes**

Table 3.1 Summary of the differences between the previous and proposed approaches for melody harmonization. The top attributes are related to the model architecture and experimental settings. The bottom attributes are related to the objectives of the studies. Bolded text indicates distinct differences between the proposed methods and the other methods.

convert a melody into a more intuitive *note-based* representation, where each frame represents one note. To this end, we use a novel *time-to-note* compression method to map a binary piano roll representation into a note-based embedding.

In addition, we expand the conventional chord prediction task to a *flexible* harmonization task using a variational autoencoder (VAE) [142]. A melody can introduce diverse interpretations from multiple perspectives toward its musical structure or the arrangers' personalities [87, 25]. Therefore, it is more intuitive to *sample* chords from the proper distribution of real-world music. Current music generation systems have also leveraged VAE-based methods to produce creative outputs from the latent space [66]. However, most previous studies of melody harmonization have aimed at the static generation of chords with fixed model parameters. Thus, we utilize the VAE setting, which explicitly approximates the general chord distribution, for stochastic harmonization.

We concretely use the *variational Transformer* inspired by Lin *et al.* [143]. They used a Transformer-based model extended by a conditional VAE framework to generate a *response* from a conditional *context*. We leverage this seq2seq architecture to achieve a variational neural machine translation (VNMT) from a given melody to the chords [144, 145, 146]. To the best of our knowledge, we are the first to apply the VNMT approach to music generation. In particular, our approach is different from previous music generation studies using the variational Transformer, which mostly served as an *autoencoder* [35, 78].

Furthermore, we attempt to regularize the variational Transformer for *controlling* the chord outputs through a disentangled representation. Generating arbitrary sets of chords may not satisfy users who would like to create music based on their own tastes. In terms of building interactive music generation systems as well as learning a good representation for sequential data, controllable generation with the VAE framework has mainly been approached by recent studies. These studies have aimed to learn disentangled representations for high-level musical features, such as pitch, rhythm, harmony, context, or arousal, through supervised learning [31, 67, 60, 38]. Inspired by these studies, we use domain-specific inductive bias to achieve a disentangled representation for the well-summarized context of the target melody and chords. In particular, we exploit an auxiliary regularization method proposed by Pati *et al.* [68] to force the target representation to be related to the musical attribute. We set the number of unique chords in a chord progression as a controllable attribute of the generated chords.

In this paper, we propose *three* Transformer-based models for structured and flexible chord generation from a given melody. These models are based on three types of Transformer architecture: 1) the **S**tandard **T**ransformer for structured **Harm**onization (**STHarm**), 2) the **V**ariational **T**ransformer for flexible **Harm**onization (**VTHarm**), and 3) the **r**egularized **V**ariational **T**ransformer for controllable **Harm**onization (**rVTHarm**). Table 3.1 summarizes additional details on how the proposed models differ from the LSTM-based approaches in terms of experimental settings and objectives. Our contribution also lies in the substantial evaluations of each model's performance using multiple datasets. One dataset is a benchmark dataset of popular music that is used for the direct comparison with previous approaches. The other dataset contains music from the contemporary genre, such as jazz, which possesses relatively higher musical tension than popular music. These datasets also differ by whether a key signature is normalized. Therefore, we assess the harmonization models in various dataset settings. The experimental results support that STHarm, VTHarm, and rVTHarm can capture structured contexts within and between melody and chord sequences, increase chord diversity, and explicitly control chord outputs, respectively, compared to LSTM-based models. The source code for the proposed methods is available at `https://github.com/rsy1026/harmonizers_transformer`.

3.2 Proposed Methods

We propose three models based on Transformer targeting structured and flexible melody harmonization. The first model uses the standard Transformer model to translate a melody to a chord sequence. The second model uses the variational Transformer to learn a global latent representation of the complete music [143]. The last model regularizes the representation of the variational Transformer to control harmonic attributes. We name these models *STHarm*, *VTHarm*, and *rVTHarm*, respectively. In each model, the Transformer encoder receives a given melody, and the decoder generates a chord sequence according to the attention weights computed between the melody and chords. The overall structures of the proposed models are illustrated in Fig. 3.1.

3.2.1 Standard Transformer Model (STHarm)

STHarm generally follows the original Transformer model, except that the input and output representations are not event-based [59]. Instead, we use a binary melody piano roll and serialized chord labels instead of musical event tokens. Each frame of the melody piano roll represents the same temporal length.

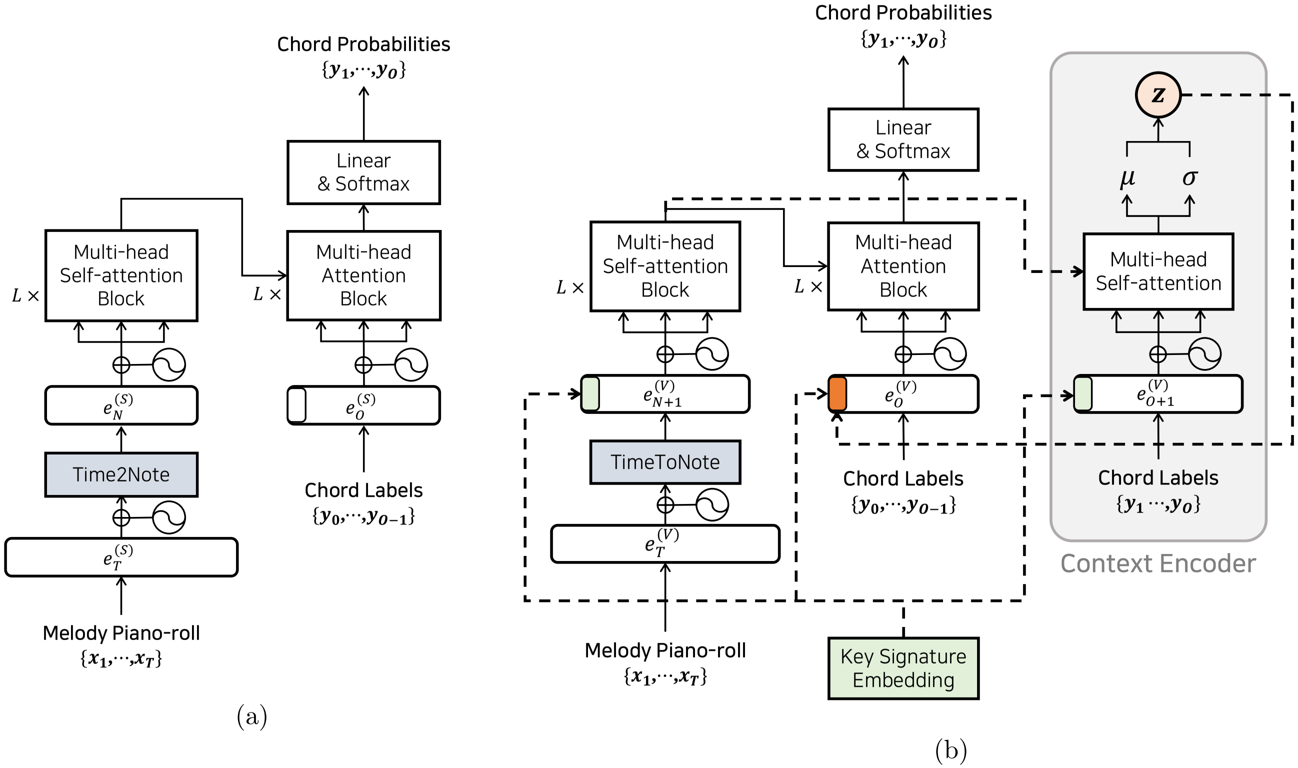


Fig. 3.1 The overall architectures of the proposed methods: (a) STHarm and (b) VTHarm and rVTHarm. VTHarm and rVTHarm share the same architecture. The colored area and dotted lines represent the modified parts from the vanilla Transformer.

Let $x_{1:T} \in \{0,1\}^{T\times|P|}$ be a one-hot vector sequence of a given melody, where T is the length of the melody, $|P|$ is the number of pitches, and t is a time index by the length of a sixteenth note. The encoder receives the input $x_{1:T}$ to capture the notewise melodic context as follows:

$$\begin{aligned} e_T^{(\mathrm{S})} &= \mathrm{Embedding}(x_{1:T}) \\ e_N^{(\mathrm{S})} &= \mathrm{TimeToNote}(e_T^{(\mathrm{S})} + w_T, M) \\ \mathrm{Enc}(x_{1:T}) &= \text{Self-AttBlocks}(e_N^{(\mathrm{S})} + w_N) \end{aligned} \tag{3.1}$$

where e_T, e_N, S, and N denote the time-level embedding vectors, note-level embedding vectors, STHarm, and the number of melody notes, respectively, Embedding and Self-AttBlocks denote the embedding layer and L multihead self-attention blocks that are identical to the vanilla Transformer, respectively [59], w_* denotes a sinusoidal positional embedding scaled by a trainable weight [147], and TimeToNote is a novel method that we propose to convert the *time-wise* embedding to the *notewise* embedding to capture the note patterns in a melody.

In the Time2Note procedure, we add the scaled positional embedding w_T to $e_T^{(\mathrm{S})}$. Then, we transfer it to the notewise embedding $e_N^{(\mathrm{S})}$ with average pooling by an alignment matrix $M \in \{0,1\}^{T\times N}$ as (3.2), where M indicates the alignment path between a piano roll and a series of notes. This process enables each frame of the notewise embedding to preserve the information of the original note duration:

$$\mathrm{TimeToNote}(e, M) = \mathrm{Linear}\left(\frac{M^{\mathrm{T}} \cdot e}{\sum_{t=1}^{T} M_{t,1:N}}\right) \tag{3.2}$$

where Linear denotes a fully connected layer. The compressed embedding $e_{1:N}^{(\mathrm{S})}$ is added to another scaled positional embedding w_N and passes through the L

multihead self-attention blocks.

The decoder receives the right-shifted target chords and computes attention with the encoder output $\mathrm{Enc}(x_{1:T})$ to predict the chords as follows:

$$\begin{aligned} e_O^{(\mathrm{S})} &= \mathrm{Embedding}(y_{0:O-1}) \\ \acute{e}_O^{(\mathrm{S})} &= \mathrm{AttBlocks}(e_O^{(\mathrm{S})} + w_O, \mathrm{Enc}(x_{1:T})) \\ p(\tilde{y}_{1:O}) &= \mathrm{Softmax}(\mathrm{Linear}(\acute{e}_{1:O}^{(\mathrm{S})})) \end{aligned} \tag{3.3}$$

where $y_{0:O-1} \in \{0,1\}^{O\times|C|}$ is a sequence of one-hot vectors for the right-shifted target chords, O is the length of the chord sequence, $|C|$ is the number of chord classes, and AttBlocks denotes L loops of the Transformer attention blocks. The final probabilities are estimated by a final linear layer with softmax activation.

3.2.2 Variational Transformer Model (VTHarm)

The proposed architecture of VTHarm is inspired by [143]. VTHarm has an additional probabilistic encoder for a latent variable z, where z represents the global attribute of the aggregated melody and chords. We denote this encoder as the *context encoder*. We add a global key signature label as a conditional input token to the model. The key signature is essential for an arbitrary melody to obtain a certain harmonic context [148]. The key signature token can aid the model in specifying the latent space and sampling the outputs from the constrained chord distributions. In contrast, STHarm does not use this token since it finds the mean distribution for chords that best fit a given melody.

The encoder used in VTHarm is identical to the encoder used in STHarm, except that the conditional token c is concatenated at the beginning of the note-based melody embedding $e_N^{(\mathrm{V})}$ as follows:

$$\begin{aligned} e_{N+1}^{(\mathrm{V})} &= \mathrm{Concat}_s(c, e_N^{(\mathrm{V})}) \\ \mathrm{Enc}(c, x_{1:T}) &= \text{Self-AttBlocks}(e_{N+1}^{(\mathrm{V})} + w_{N+1}) \end{aligned} \tag{3.4}$$

where Concat_s denotes the concatenation over the sequence dimension. The self-attention block can connect c and the remaining parts of the embedding and convey any constraints to the whole embedding.

The context encoder infers the latent representation z from the encoder output, chord input y, and conditional token c as follows:

$$\begin{aligned} e_{O+1}^{(\mathrm{V})} &= \mathrm{Concat}_s(c, \mathrm{Embedding}(y_{1:O})) \\ \acute{e}_{O+1}^{(\mathrm{V})} &= \text{Self-AttBlock}(e_{O+1}^{(\mathrm{V})} + w_{O+1}) \\ r &= \mathrm{Concat}_d(\mathrm{Pool}(\mathrm{Enc}(c, x_{1:T})), \mathrm{Pool}(\acute{e}_{O+1}^{(\mathrm{V})})) \\ [\mu, \sigma] &= \mathrm{Linear}(r) \qquad\qquad z \sim \mathcal{N}(\mu, \sigma) \end{aligned} \tag{3.5}$$

where V denotes VTHarm, Concat_d denotes the concatenation over the feature dimension, Pool denotes the average pooling over time, and self-AttBlock denotes only one loop of the self-attention block. The context encoder maps the chord input $y_{1:O}$ into the embedding $e_O^{(\mathrm{V})}$. Then, c is concatenated at the beginning of $e_O^{(\mathrm{V})}$ over the sequence dimension before the multihead self-attention blocks. The self-attention output contains the harmonic context according to the key information. It is mean-aggregated over time so that it represents the global information of the chords [35]. The encoder output $E(c, x_{1:T})$ is also mean aggregated over time to represent the global attribute of a melody. These two aggregated vectors are concatenated over the feature dimension and pass through the bottleneck, resulting in two parameters, μ, and σ. The latent code z is inferred from μ and σ through the reparameterization trick, and its prior is assumed to be the normal distribution [142].

The decoder reconstructs the target chords from the right-shifted chord input and encoder output, conditioned by c and the latent variable z as follows:

$$\begin{aligned} e_o^{(\mathrm{V})} &= \mathrm{Concat}_s(z + c, \mathrm{Embedding}(y_{1:O-1})) \\ \acute{e}_O^{(\mathrm{V})} &= \mathrm{AttBlocks}(e_O^{(\mathrm{V})} + w_O, \mathrm{Enc}(x_{1:T})) \\ p(\tilde{y}_{1:O}) &= \mathrm{Softmax}(\mathrm{Linear}(\acute{e}_O^{(\mathrm{V})})) \end{aligned} \tag{3.6}$$

The right-shifted chord input is first encoded with the same lookup table from the context encoder. The latent variable z and the key signature token c are added to the beginning, which corresponds to the “start-of-sequence" part of the chord embedding. The following attention network transfers the aggregated information from z and c to all frames of the embedding. The rest of the Transformer decoder reconstructs the target chords.

3.2.3 Regularized Variational Transformer Model (rVTHarm)

Training VTHarm alone cannot guarantee a disentangled representation of the desired aspect. Therefore, rVTHarm aims to achieve a disentangled representation to control the generated chord outputs. We use the auxiliary loss by Pati *et al.* [68] to directly supervise the latent representation z. In this study, we choose the number of unique chords in the progression, or *chord coverage*, as a naive attribute for the chord complexity [25].

The regularization function from Pati *et al.* assumes that the target dimension of the latent representation can be disentangled by its monotonic relationship with a specific attribute [68]. For example, the target attribute value should increase when the constrained latent dimension is modulated toward a positive direction. To this end, the difference between the attribute values of an arbitrary pair of two samples is forced to be the same sign as that between the corresponding latent representations. Let a_i and a_j be the target attribute values of the ith and jth batches, respectively, where $i, j \in [1, B]$ and B is the batch size. Similarly, let z_i^r and z_j^r be the rth dimension values of the latent variables of the ith and jth batches, respectively. A distance matrix $\mathcal{D}_r$ is computed between all pairs of z_i^r and z_j^r in the mini-batch. The corresponding $\mathcal{D}_a$ is computed in the same way between all pairs of a_i and a_j. We minimize the difference between $\mathcal{D}_r$ and $\mathcal{D}_a$ as follows:

$$\mathcal{L}_{\text{Reg}} = \text{MSE}(\tanh(\mathcal{D}_r), \text{sign}(\mathcal{D}_a)) \tag{3.7}$$

where MSE is the mean squared error. In this paper, we regularize the first dimension of z, so $r = 1$.

3.2.4 Training Objectives

The main objective for STHarm is maximizing the log likelihood of the estimated chord sequence y given the melody x:

$$\mathcal{L}_{\text{ST}} = \mathbb{E}[-\log p_\theta(y|x)] \tag{3.8}$$

where θ are the model parameters of STHarm.

In VTHarm, the main goal is to approximate the marginal distribution of y through the objective of negative evidence lower bound (ELBO) by minimizing the losses for the reconstruction and Kullback-Leibler divergence (KLD) [142]. The chord probability $p_\theta(y)$ and posterior distribution $q_\phi(z)$ are conditioned by the melody input x and key signature token c, whereas the prior $p_\theta(z)$ is the normal distribution following the conditional VAE framework [149]:

$$\begin{aligned}\mathcal{L}_{\text{VT}} &= \mathbb{E}_{q_\phi(z|x,y,c)}[-\log p_\theta(y|x,z,c)] \\ &+ \lambda_{KL}\text{KL}(q_\phi(z|x,y,c)\|p_\theta(z))\end{aligned} \tag{3.9}$$

where q_ϕ is the posterior distribution of z parameterized by ϕ, and λ_{KL} is a hyperparameter for balancing the KLD loss term [107, 66].

This training objective is expanded in rVTHarm by the explicit regularization of the latent space. Therefore, rVTHarm shares the overall objective with VTHarm except for the added regularization term as follows:

$$\mathcal{L}_{\text{rVT}} = \mathcal{L}_{\text{VT}} + \lambda_{Reg}\mathcal{L}_{\text{Reg}} \tag{3.10}$$

where λ_{Reg} is a hyperparameter for balancing the auxiliary loss term.

To generate chords, VTHarm and rVTHarm autoregressively sample the chord output $y_{1:O}$ from the melody input $x_{1:T}$, latent variable z, and conditional token c as follows:

$$p_\theta(y|x,z,c) = \prod_{O} p_\theta(y_o|x_{1:t}, y_{0:o-1}, z, c) \tag{3.11}$$

where z is sampled from the normal prior $\mathcal{N}(0,1)$.

3.3 Experimental Settings

We conduct objective and subjective evaluations for the three proposed methods. In this section, we explain the settings for the corresponding experiments. We first introduce the two datasets used for the experiments. Next, we summarize the baseline models, model settings, and metrics for the evaluations.

3.3.1 Datasets

We set $|P| = 13$ for the 12 pitch classes and rest. We convert all chords into one of the 72 chords, which are triad chords in major, minor, diminished, and seventh chords in major, minor, dominant, so that $|C| = 72$. Each note and chord are quantized by lengths of sixteenth note and a half-measure for all datasets, respectively. The length of each batch is a maximum of 8 measures. We only use songs with a time signature of 4/4 and all songs are set to 120 BPM. The training, validation, and test sets for each dataset are divided into approximately an 8:1:1 ratio. We construct batches by slicing each song into excerpts of 8-measures where 2-measures overlap. For each test, we extract 8-measure excerpts without an overlap. We use two public datasets that differ in some experimental settings as well as musical characteristics: the Chord Melody Dataset (CMD) and the Hooktheory Lead Sheet Dataset (HLSD).

The Chord Melody Dataset (CMD). CMD [140] is composed of 473 songs in contemporary genres such as jazz and pop. The songs in this database are only in the major key, and most of them are transposed to all 12 keys. We choose this dataset to examine the model performance from the complex chords in various keys with nontrivial tensions. The lead sheets are in the music extensible markup language (MusicXML) format, where the melody and chord labels are manually annotated and are parsed with the existing MusicXML parser [150, 47]. We use 389 songs for the training set and the rest for the validation and test sets (48 songs each). As a result, we use 36,528, 1,756, and 165 samples for the training, validation, and test sets, respectively.

The Hooktheory Lead Sheet Dataset (HLSD). HLSD [139] is an online database of melody and chord annotations that cover various genres, such as the pop, new age, and original soundtracks. This dataset has been constructed on a crowdsourcing platform called TheoryTab [1], in which users have transcribed a large number of high quality melodies and chords. This dataset contains the raw annotations of melodies and chords in XML format, JSON data of the symbolic features of melodies and chords, and piano-roll figures depicting the melody and chords. We use the JSON data for 9,218 songs divided into 13,335 parts. We also normalize all songs into C major or C minor, as in previous studies [25, 91]. Following Sun *et al.* [91], we use 500 parts for the test set and the other 500 parts for the validation set. As a result, we use 32,619, 1,346, and 809 samples for the training, validation, and test sets, respectively.

3.3.2 Comparative Methods

We use two baseline models and one ground truth for our study. **BLSTM** by Lim *et al.* [26] is composed of two stacked layers of bidirectional LSTM. This model has been a base for most of the recent deep learning approaches [25, 91]. We use BLSTM to compare the stacked RNN structure with Transformer. **ONADE** by Sun *et al.* [91] uses the orderless NADE and Gibbs sampling. This model represents a BLSTM-based model with randomness and improved chord diversity. For the ground truth, we use the original progressions from the datasets. We denote the ground truth as **Human**.

3.3.3 Training

The embedding sizes of the melody and chord are 128 and 256, respectively. We use a hidden size of 256, attention head size of 4, number of attention blocks L

[1] https://www.hooktheory.com/theorytab

of 4, and size of the latent variable z of 16. A dropout layer is used after every scaled positional encoding at a rate of 0.2. We use an Adam optimizer [151] with an initial learning rate of 1e-4, which is reduced to 95% after every epoch. We train the proposed models for 100 epochs with a batch size of 128. To select the value of λ_{KL}, we refer to several studies on VAE-based music generation in which a scaling weight smaller than 1 encourages better reconstruction [66, 21]. Then, we empirically set λ_{KL} and λ_{Reg} to be 0.1 and 1, respectively, which results in the best performance.

The models are implemented and evaluated in Python 3 and the PyTorch deep learning framework of version 1.5.0. For training each model, we use one NVIDIA GeForce GTX 1080 Ti. We mostly refer to the previous implementations [152, 147] when implementing the vanilla Transformer. For implementing and training BLSTM and ONADE, we use the original settings [26, 91]. The gradients are all clipped to 1 for the learning stability during training of all models. VTHarm, rVTHarm, and ONADE are assessed with 10 test samples per melody due to their randomness. Other models are evaluated with the samples in maximum probabilities. We use the truncation trick with a threshold of 3 for VTHarm and rVTHarm in qualitative and subjective tests [153].

3.3.4 Metrics

We introduce three categories of metrics for evaluating the proposed models: chord coherence and diversity, harmonic similarity, and subjective evaluation.

Chord Coherence and Diversity

We use six canonical metrics proposed by Yeh *et al.* that have been leveraged by recent studies [25, 91]. In brief, **chord histogram entropy (CHE)** and **chord coverage (CC)** measure chord diversity. **Chord tonal distance (CTD)** measures the coherence of the chord transition. **Chord tone to non-chord tone ratio (CTR)**, **pitch consonance score (PCS)**, and **melody-chord tonal distance (MTD)** measure the coherence between the melody and chords:

- **Chord histogram entropy (CHE).** This metric computes the entropy from the histogram of $|C|$ bins that counts the occurrences of the chord classes within the chord sequence:

$$\mathrm{CHE} = -\sum_{i=1}^{|\mathcal{C}|} p_i \log p_i \tag{3.12}$$

  where p_i denotes the probability of the ith bin of the histogram.

- **Chord coverage (CC).** This metric is the number of unique chord labels that occur in the chord sequence.

- **Chord tonal distance (CTD).** This metric is the Euclidean distance between two 6-D tonal centroid vectors that respectively represent the two adjacent chords. Each tonal centroid vector $\zeta_n(d)$ is calculated from the pitch class profile (PCP) features as follows [154, 155]:

$$\zeta_n(d) = \frac{1}{\|\mathbf{c}_n\|_1} \sum_{l=0}^{11} \Phi(d,l)\mathbf{c}_n(l) \qquad \begin{matrix} 0 \leq d \leq 5 \\ 0 \leq l \leq 11 \end{matrix} \tag{3.13}$$

  where n is the chord index, d is one of the dimension indices of the 6-D tonal space, $\mathbf{c}_n$ is the PCP vector of the nth chord, where the number of entries for the chord tones is 1 ($\mathbf{c}_n \in \{0, 1\}$), l denotes one of the 12 entries of the PCP vectors, where each entry corresponds to each pitch class, and $\phi(d, l)$ denotes the dth basis of the 6-D tonal space for the lth entry of the PCP vector. Each basis is defined as follows:

$$\phi_l = \begin{bmatrix} \Phi(0,l) \\ \Phi(1,l) \\ \Phi(2,l) \\ \Phi(3,l) \\ \Phi(4,l) \\ \Phi(5,l) \end{bmatrix} = \begin{bmatrix} r_1 \sin l\frac{7\pi}{6} \\ r_1 \cos l\frac{7\pi}{6} \\ r_2 \sin l\frac{3\pi}{2} \\ r_2 \cos l\frac{3\pi}{2} \\ r_3 \sin l\frac{2\pi}{3} \\ r_3 \cos l\frac{2\pi}{3} \end{bmatrix} \quad 0 \leq l \leq 11 \tag{3.14}$$

where ϕ_l is the complete transition matrix of the 6-D feature vector for the lth entry of the PCP vector, r_1, r_2 and r_3 are the radii of the three circles that represent the 6-D tonal space. They are set to 1, 1, and 0.5, respectively, as in Harte *et al.* [155]. We compute the average of the CTD values for all pairs of adjacent chords in each progression.

- **Chord tone to non-chord tone ratio (CTR).** Originally named CT-nCTR, this metric is the ratio of the number of chord tones compared to the number of nonchord tones and *proper* nonchord tones, which have a maximum of 2-semitone intervals to the right-after note:

$$\text{CTR} = \frac{n_\text{c} + n_\text{p}}{n_\text{c} + n_\text{n}} \tag{3.15}$$

where n_c, n_n, and n_p denote the number of chord tones, nonchord tones, and proper nonchord tones, respectively, that are computed from the melody notes and corresponding chord labels.

- **Pitch consonance score (PCS).** This metric is a consonance score based on pitch intervals between the melody note and corresponding chord notes. The pitches of the melody notes are assumed to always be higher than those of the chord notes. According to the pitch interval, PCS is one of {-1, 0, 1}: 1 for perfect 1st and 5th, major/minor 3rd and 6th; 0 for perfect 4th; and -1 for other intervals. The PCS values within each sixteenth-note window are aggregated into the average. We compute the total average of the aggregated PCS for all windows over time.

- **Melody-chord tonal distance (MTD).** Originally named MCTD, this metric is the tonal distance between each melody note and its corresponding chord label. It is calculated in the same way as CTD. Each MTD value is weighted by the duration of the corresponding melody note. We average the MTD values for all of the melody notes and their chord labels.

Harmonic Similarity

We measure the similarity between the generated and human-composed chords with three metrics and assume that the chord progressions in the human-composed music inherit hierarchical and metrical structures [141, 156]. Hence, we set the human-composed music as the ground truths of the structured harmonization. Concretely, a system that generates chord progressions *similar* to human-composed music is assumed to achieve more structured harmonization [90].

Briefly, **the Levenshtein edit distance (LD)** is the global matching score between two chord sequences. **The tonal pitch step distance (TPSD)** and **directed interval class distance (DICD)** measure the distance between two chord progressions:

- **Levenshtein edit distance (LD).** LD is the Levenshtein edit distance between the generated chord labels and the ground-truth labels [90]. It measures the extent to which the generated chords are substituted for human-composed chords.

- **Tonal pitch step distance (TPSD).** TPSD computes the geometrical dissimilarity between the generated chords and the ground-truth chords in terms of the tonal pitch space (TPS) chord distance rule [157]. The TPS between chord x and chord y is computed as follows:

  $$\mathrm{TPS}(x, y) = j + k \tag{3.16}$$

  where j is the least number of steps in one direction from the chordal root of x to that of y according to the circle-of-fifths rule. In the circle-of-fifths rule, all pitch classes are arranged in intervals of either perfect fifth or fourth [158]. The variable k is the number of unique pitch class indices in the four levels (root, fifths, triadic, diatonic) within the basic space of y compared to x [157]. That is, if the pitch class index is shared by y and x, it is not counted. We compute the TPS values between all pairs of adjacent chords within each progression, resulting in a step function. TPSD is calculated as the area between the two step functions derived from the two chord progressions.

- **Directed interval class distance (DICD).** DICD computes the city block distance between the directed interval class (DIC) representation vectors for the chord transitions [159]. DIC is the histogram vector of the directional pitch interval classes, ranging from -5 to 6, computed between all pairs of chord notes from the two adjacent chords. We calculate each pitch interval from each note of the first chord to all notes of the second chord. DICD indicates both the tonal distance and *direction* between the two successive chords.

Subjective Evaluation

We expand the conventional criteria [25, 91] for deeper analysis of human judgment. **Harmonicity** measures how coherent the chords are with a given melody. **Unexpectedness** measures how much the chords deviate from expectation. **Complexity** measures how complex chord progression is perceived to be. **Preference** measures personal favor for chord progression [26].

3.4 Evaluation

In this section, we introduce the experimental results of the objective and subjective evaluations into several categories as follows. First, we compare the results of the proposed models in **chord coherence and diversity** with the baseline models. Next, we measure **harmonic similarity to human-composed music** for all models to examine whether the proposed models can result in structured harmonization. Then, we check with the **controllability of rVTHarm** for the intended factor compared with VTHarm. In addition, we introduce the results for the **subjective evaluation** and discuss the corresponding results. Moreover, we illustrate some **qualitative results** for all models to verify the strength of the proposed model. Last, we show an **ablation study** to investigate the influence of the information of the key signature

Dataset	Chord Melody Dataset					
Category	Diversity		Coherence			
Metric	CHE↑	CC↑	CTD↓	CTR↑	PCS↑	MTD↓
BLSTM	1.380	5.297	0.497	1.170	**0.543**	**1.302**
ONADE	1.389	5.482	0.502	1.220	0.497	1.362
STHarm	1.349	5.030	**0.443**	1.213	0.428	1.396
VTHarm	**1.877**	**7.523**	0.631	1.225	0.374	1.428
rVTHarm	1.705	6.202	0.508	**1.227**	0.394	1.419
Human	1.618	6.412	0.580	1.301	0.389	1.408
Dataset	Hooktheory Lead Sheet Dataset					
Category	Diversity		Coherence			
Metric	CHE↑	CC↑	CTD↓	CTR↑	PCS↑	MTD↓
BLSTM	0.928	3.262	0.609	1.146	**0.639**	**1.328**
ONADE	1.123	4.243	0.467	1.136	0.470	1.392
STHarm	0.994	3.193	**0.446**	**1.150**	0.522	1.396
VTHarm	**1.543**	**5.356**	0.696	1.147	0.459	1.435
rVTHarm	1.440	4.678	0.536	1.146	0.445	1.447
Human	1.356	4.686	0.626	1.180	0.497	1.400

Table 3.2 Evaluation results for chord coherence and diversity. CHE and CC measure the chord diversity, whereas the remaining four metrics measure the chord coherence: CTD measures the coherence of the chord progression itself. CTD, CTR, PCS, and MTD measure how harmonic the chord progression is with the given melody.

added to the variational models.

3.4.1 Chord Coherence and Diversity

We evaluate the overall coherence and diversity of the generated chords. Table 3.2 shows the results for all models. VTHarm and rVTHarm show higher CHE and CC than the baseline models in both datasets. This result indicates that these models have higher chord diversity than the baseline models. STHarm, on the other hand, reveals the lowest CTD and the lowest CHE and CC for all datasets except for CHE on HLSD. This implies that STHarm can generate smoother and simpler chord transitions than other models [91]. BLSTM and ONADE show better PCS and MTD but lower chord diversity than the proposed models.

Meanwhile, Human shows worse scores for chord coherence than STHarm for the following reasons. 1) The human-composed samples from CMD and HLSD include 72 different chord types with various amounts of musical tensions. 2) STHarm may generate common chords more frequently from the average chord distribution than the human-composed music, as shown in the lower diversity scores. Concretely, the most frequent chords in real-world music are diatonic chords such as the C, G, and F major chords in the C major key [26]. Since these chords have relatively less musical tension with respect to a melody, they are close to the melody under a music-theoretical space. Thus, these chords may obtain better coherence scores than other chords with more musical tension.

Moreover, Human shows lower diversity scores than the variational models. We assume that this is because these models can produce some infrequent chords far from the mean distribution of real-world music. The nature of stochastic generation models draws samples from the normal distribution [153]. Some of the generated chords may violate the given key signature but increase the information outside the certain harmonic context. Hence, they may contribute to higher chord diversity than human-composed music.

Consequently, the overall results reflect a trade-off between chord coherence and diversity [88, 25]. Additionally, Human cannot serve as the upper bound for the six metrics in both datasets. Therefore, these metrics cannot function as complete criteria for determining the *good* harmonization but only show the model tendencies in the music-theoretical perspective [25, 91]. Hence, we are inspired to use additional criteria to evaluate the generated outputs with respect

Dataset	Chord Melody Dataset		
Metric	LD↓	TPSD↓	DICD↓
BLSTM	**0.75(±0.20)**	2.63(±1.11)	116.45(±42.98)
ONADE	0.85(±0.17)	2.80(±1.06)	128.10(±41.51)
STHarm	0.80(±0.21)	**2.43(±1.35)**	**107.68(±44.86)**
VTHarm	0.86(±0.14)	2.72(±1.02)	121.08(±36.82)
rVTHarm	0.86(±0.15)	2.71(±1.17)	118.01(±36.59)
Dataset	Hooktheory Lead Sheet Dataset		
Metric	LD↓	TPSD↓	DICD↓
BLSTM	**0.62(±0.21)**	2.48(±1.11)	85.39(±35.22)
ONADE	0.90(±0.14)	2.75(±1.17)	116.16(±39.67)
STHarm	0.65(±0.25)	**2.17(±1.55)**	**75.54(±40.81)**
VTHarm	0.77(±0.16)	2.54(±1.15)	98.55(±33.53)
rVTHarm	0.79(±0.16)	2.32(±1.26)	91.63(±34.92)

Table 3.3 Evaluation results for the chord similarity metrics. Lower scores correspond to higher human composition similarity.

to human-composed chords.

3.4.2 Harmonic Similarity to Human

We investigate the harmonic similarity between the human-composed and generated chords. We use the samples from Human as the ground truth. This explicit comparison with Human can provide insight into whether the generated chords from each model are as well-structured as human-composed music [56].

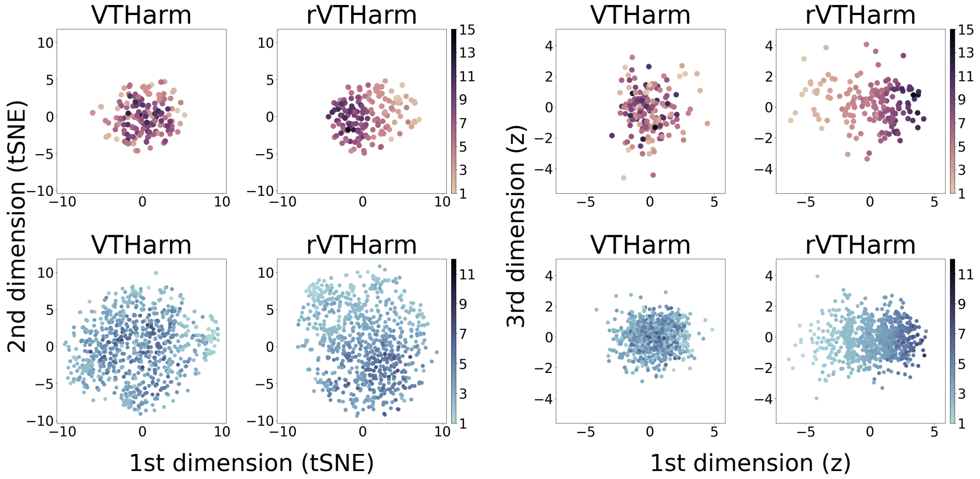
The harmonic similarity results are shown in Table 3.3. BLSTM shows the lowest LD compared to the proposed models, whereas ONADE shows the highest LD in all datasets. This indicates that BLSTM is better than the proposed models at providing the right chords to the melody. However, the better matching of individual chords does not correspond to the higher similarity of the chord sequence in terms of musical structure [157].

For TPSD and DICD, STHarm shows the lowest scores in all datasets. This implies that STHarm can generate chord patterns that is more similar to Human than other models. VTHarm and rVTHarm show higher LD scores than BLSTM but better similarity scores than ONADE. This indicates that the VT models tend to have higher substitution probabilities between chords than BLSTM [157]. This is possible because the VT models are trained to induce some infrequent chords that are far from the mean distribution of real-world chords. Nonetheless, the VT models are better than ONADE at creating more human-like chord patterns, even with a larger variety of chord types. Moreover, rVTHarm shows better TPSD and DICD scores than VTHarm in both dataset. It implies that explicit regularization of the latent representation can encourage the model to generate structured chords.

3.4.3 Controlling Chord Complexity

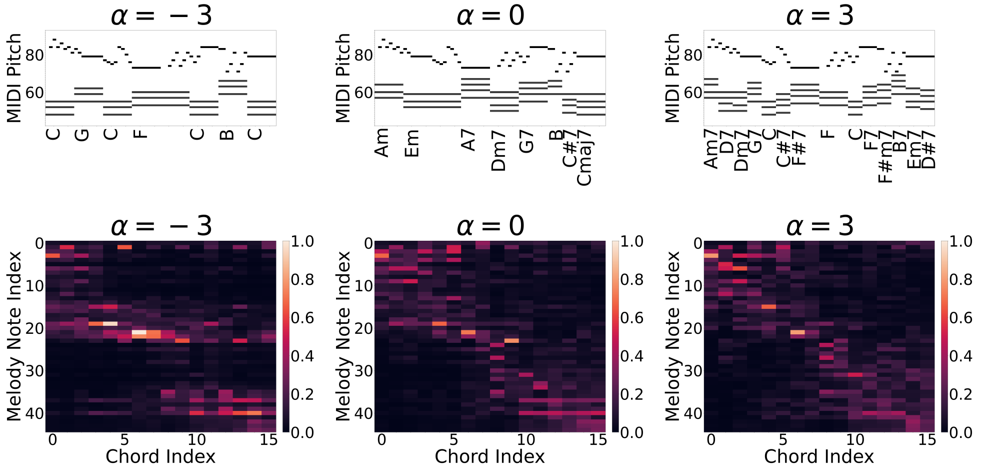
We verify the monotonic relationship between the chord attribute and z from rVTHarm. We use VTHarm and rVTHarm to infer z from the test melodies and chords. Then, the dimension of z is reduced by two with t-stochastic neighbor embedding (tSNE) [60]. When visualizing, we use the chord coverage value as the third dimension (hue). The tSNE results and two dimensions, the first and third, of the original z are illustrated in Fig. 3.2. This figure shows that the tSNE results of rVTHarm are grouped by the attribute compared to VTHarm. The first dimension of z from rVTHarm is also shown to be monotonically related to the attribute [68].

In addition, we examine the attention maps of rVTHarm with different values of α. We randomly sample z, where α is set to be one of $\{-3, 0, 3\}$, and



(a) (b)

Fig. 3.2 Visualization of (a) tSNE results and (b) two dimension values from z. The top (purple) and bottom (indigo) rows represent the CMD and HLSD, respectively. The hue of each plot represents the chord coverage value.



(a) (b) (c)

Fig. 3.3 The generated results from rVTHarm in the piano-rolls (top) and the corresponding attention matrices (bottom). (a), (b), and (c) represent the results from different values of $a \in \{-3, 0, 3\}$.

Dataset	CMD	HLSD
VTHarm	-0.1132	0.0805
rVTHarm	**0.5332**	**0.4512**

Table 3.4 Pearson's correlation coefficients between α and CC of the generated outputs from VTHarm and rVTHarm. CMD and HLSD are the Chord Melody Dataset and Hooktheory Lead Sheet Dataset, respectively.

generate the chords from z and the test melodies. We sum the attention matrices along the head dimension to see the aggregated weights. Fig. 3.3 shows that the attention weights become balanced and diagonal when α increases from -3 to 3. This implies that the decoder of rVTHarm tends to focus on more melody notes when α increases.

Furthermore, we compute Pearson's correlation coefficients between α and the CC scores of the corresponding chord outputs. Table 3.4 shows that rVTHarm reveals higher correlation coefficients than VTHarm for all datasets. This confirms that rVTHarm derives a meaningful representation for the intended chord attribute compared to VTHarm.

3.4.4 Subjective Evaluation

We conduct a listening test for subjective evaluation. We extract the samples in 8-measure length from the arbitrary parts of each melody. For rVTHarm, we sample z by setting a to randomly be $\{-3, 0, 3\}$. The listening test comprises ten trials, where each trial contains six samples of all comparative methods for one melody. A participant[2] grades four metrics, Harmonicity (H), Unexpectedness (U), Complexity (C), and Preference (P), on a five-point Likert scale

[2] Every experimental protocol was approved by the Institutional Review Board (IRB) of Seoul National University. Written consent forms were collected from the participants, and the study was conducted according to the ethical standards outlined in the 1962 Helsinki Declaration.

Condition	*With* Melody Awareness			
Metric	H	U	C	P
BLSTM	3.29(±1.00)	2.67(±1.04)	2.42(±0.98)	2.88(±1.11)
ONADE	2.91(±1.03)	2.98(±1.07)	2.89(±1.01)	2.69(±1.07)
STHarm	**3.44(±1.01)**	2.33(±0.99)	2.33(±1.08)	3.11(±1.16)
VTHarm	2.95(±1.04)	**3.23(±1.01)**	3.05(±0.98)	2.83(±1.07)
rVTHarm($\alpha = -3$**)**	3.02(±1.27)	2.98(±1.00)	2.56(±0.80)	2.67(±1.23)
rVTHarm($\alpha = 0$**)**	3.33(±1.02)	2.72(±0.97)	2.72(±0.90)	**3.17(±1.26)**
rVTHarm($\alpha = 3$**)**	3.20(±1.04)	3.17(±0.93)	**3.44(±0.88)**	3.10(±1.16)
Human	3.41(±1.13)	2.93(±1.05)	2.92(±1.00)	3.33(±1.17)

Condition	*Without* Melody Awareness			
Metric	H	U	C	P
BLSTM	2.87(±1.05)	2.96(±1.02)	2.68(±0.95)	2.51(±1.10)
ONADE	2.76(±1.03)	3.09(±1.01)	2.90(±1.05)	2.57(±1.12)
STHarm	**3.20(±1.15)**	2.68(±1.00)	2.65(±1.01)	**2.92(±1.18)**
VTHarm	2.87(±1.11)	3.18(±1.02)	2.97(±0.95)	2.65(±1.06)
rVTHarm($\alpha = -3$**)**	2.94(±1.14)	2.72(±1.00)	2.38(±1.02)	2.53(±1.13)
rVTHarm($\alpha = 0$**)**	2.90(±1.01)	2.99(±1.00)	3.01(±0.97)	2.57(±0.99)
rVTHarm($\alpha = 3$**)**	2.56(±1.03)	**3.56(±0.98)**	**3.28(±0.98)**	2.42(±1.02)
Human	3.15(±1.15)	2.96(±1.04)	2.97(±1.08)	3.00(±1.19)

Table 3.5 Subjective evaluation results for the six methods according to whether the participants have known the given melody.

for each method [25, 91]. We denote these metrics as “H", “U", “C", and “P" for simplicity. We collect answers on whether a participant is familiar with a given melody as in Lim *et al.* [26]. A total of 37 participants were involved in the listening test: 3 participants had degrees in music. Thirty-two participants indicated that they had musical backgrounds, and 25 participants mentioned that they usually listened to popular music.

Table 3.5 shows that the results mainly support the quantitative evaluation results. We report rVTHarm according to different values of α to investigate the affect of intentionally controlled chord complexity. In particular, STHarm shows the highest H and P scores, except that it receives the second-highest P score from the participants with melody awareness. This suggests that STHarm tend to output more plausible chords to listen to than the baseline models. For U and C, VTHarm and rVTHarm with three settings of α generally show higher scores than the other models. On the other hand, these variational models show lower harmonicity and preference scores than STHarm in most cases. We assume that the variational models tend to generate more chords far from the mean distribution of the learned music data than STHarm. Such unique chords can reveal more inharmonicity than the frequent chords, and it may have provided the participants with unpleasant feelings. In addition, most participants listened to popular music, where common chords with less musical tension are used. Therefore, it may have led the participants providing poorer scores on preference as well as harmonicity. Nevertheless, VTHarm and rVTHarm mostly show better H and P scores than ONADE which is a comparable model with similar U and C scores. It means that the outputs from VTHarm and rVTHarm are more plausible and persuasive than the baseline model with similar unexpectedness and complexity.

We also analyze the subjective results according to melody awareness. The results for the two-way analysis of variance (ANOVA) show that melody awareness and method type significantly affect all metric scores ($p < 0.05$). All models achieve higher P scores with melody awareness. In particular, VTHarm and rVTHarm ($\alpha >= 0$) show higher H and P scores than ONADE, and rVTHarm ($\alpha >= 0$) especially outperforms BLSTM in H and P. This implies that the samples from the VT models in moderate or high chord complexity are likely to be perceived as more plausible than the baseline models by participants who know the given melodies. Comparing VTHarm and rVTHarm, rVTHarm shows higher H and P scores than VTHarm except for the P score from rVTHarm

Model	H	B	O	S	V	R(-3)	R(0)	R(3)
P(aware)	-0.17	-0.09	-0.04	0.01	-0.08	-0.20	-0.28	**-0.46**
C(aware)	0.47	0.65	0.56	0.49	0.53	**0.02**	0.38	0.41
P(unaware)	-0.21	-0.22	-0.03	-0.10	-0.16	-0.21	0.19	-0.11
C(unaware)	0.56	0.47	0.45	0.60	0.44	0.44	0.49	0.30

Table 3.6 Pearson's correlation coefficients of U score with P and C scores for Human (H), BLSTM (B), ONADE (O), STHarm (S), VTHarm (V), and rVTHarm ($\alpha = n$) (R(n)) according to the melody awareness.

($\alpha = -3$). In particular, rVTHarm with $\alpha = 0$ shows the best P score and the second-highest H score among the other models. It indicates that rVTHarm with average chord coverage can generate chords that are preferable and unexpected at the same time under the known melody compared to other models. In other words, regularizing the latent representation may be beneficial to increase generation power for the desired harmonization of the given melodies.

When the melody is unaware, BLSTM and rVTHarm obtain significantly lower P scores compared to when the melody is aware ($p < 0.001$). We further compute Pearson's correlation coefficient of U with C or P scores, as shown in Table 3.6. As a result, rVTHarm reveal strong negative correlations of U with both C and P scores when the melody is aware, compared to VTHarm which shows the highest U score with melody awareness. On the other hand, rVTHarm with $\alpha = -3$ shows the smallest correlation between U and C. This indicates that 1) the unexpected chords generated from rVTHarm may harm their preference to a larger extent than VTHarm when the melody is known, and 2) some factors other than complexity seem to cause the unexpectedness in rVTHarm with $\alpha = -3$. However, the mean preference scores of rVTHarm significantly increase with melody awareness compared to those without melody awareness. One of the presumptions is that the familiarity of the melody may

Condition	*With* Melody Awareness			
Metric	r_{H}	r_{U}	r_{C}	r_{P}
rVTHarm$(\alpha = -3)$	**-0.2806**	0.0040	-0.0286	**-0.1823**
rVTHarm$(\alpha = 0)$	-0.2768	0.1391	0.0320	-0.1337
rVTHarm$(\alpha = 3)$	-0.1173	**0.1977**	**0.2598**	-0.0299
Condition	*Without* Melody Awareness			
Metric	r_{H}	r_{U}	r_{C}	r_{P}
rVTHarm$(\alpha = -3)$	-0.0756	**0.0750**	**0.1505**	-0.0955
rVTHarm$(\alpha = 0)$	0.0869	-0.1023	-0.0961	0.0887
rVTHarm$(\alpha = 3)$	**-0.2021**	-0.0011	0.0331	**-0.0992**

Table 3.7 Pearson's correlation coefficients between the scores for each metric in rVTHarm ($\alpha \in \{-3, 0, 3\}$) and the mismatch among α and the original chord coverage values of the test samples. $r*$ denotes the correlation coefficient with respect to each metric. Bolded values denote those with the largest magnitude among the three models.

strongly compensate for the high unexpectedness of the chords generated by rVTHarm.

The main difference between rVTHarm and the other proposed models is that the generated chords are arbitrarily controlled by chord coverage, regardless of the harmonic rhythm inherited by the given melodies. Therefore, melody awareness by the participants can influence their perception of some mismatches between the original chord coverage for the melody and the controlled chord coverage. We investigate whether such mismatches can actually affect the metric scores of the listening test. To this end, we additionally compute Pearson's correlation coefficient between each metric score and the mismatch in chord coverage. Each mismatch is defined as the absolute difference between α and the chord coverage values normalized into $[-3, 3]$.

According to Table 3.7, knowing the melody generally derives a larger magnitude of the correlation coefficients than not knowing the melody. Concretely, the participants who know the melody are more likely to perceive decreasing

H and P for the generated chords when mismatches in chord complexity are increasing. In contrast, the participants can recognize higher U and C when the mismatches get larger. Furthermore, when $\alpha = -3$, the correlation is the most negative for H and P, while it gets the most positive in U and C scores when $\alpha = 3$. It suggests that H and P become lower when the chord coverage is lower than the original, while U and C become higher when the chord coverage is higher than the original. On the other hand, the participants without melody awareness are less likely to be affected by the mismatches in chord coverage than those with melody awareness. Interestingly, a tendency of the results is almost reversed from that with melody awareness: rVTHarm with $\alpha = 3$ shows the most negative correlations for H and P, while $\alpha = -3$ leads to the most positive correlations for U and C. Moreover, rVTHarm in average chord coverage shows opposite directions of correlation from other models. Overall, these results reveal the significant influence of melody awareness on the evaluation metrics. Hence, these phenomena need to be deeply investigated in the future to improve the robustness of controllable melody harmonization.

3.4.5 Qualitative Results

Figs. 3.4 and 3.5 show some of the actual samples from the listening test for all five models as well as the human-composed music. These samples reveal the strengths of the proposed models. First, Fig. 3.4 mainly shows that the proposed models tend to reproduce the binary metrical structure of the chords compared to the baseline models. The binary metric structure is close to real-world music, most of which has been composed of four beats and strongly influenced by metrical boundaries [156]. In contrast, the chords generated from the baseline

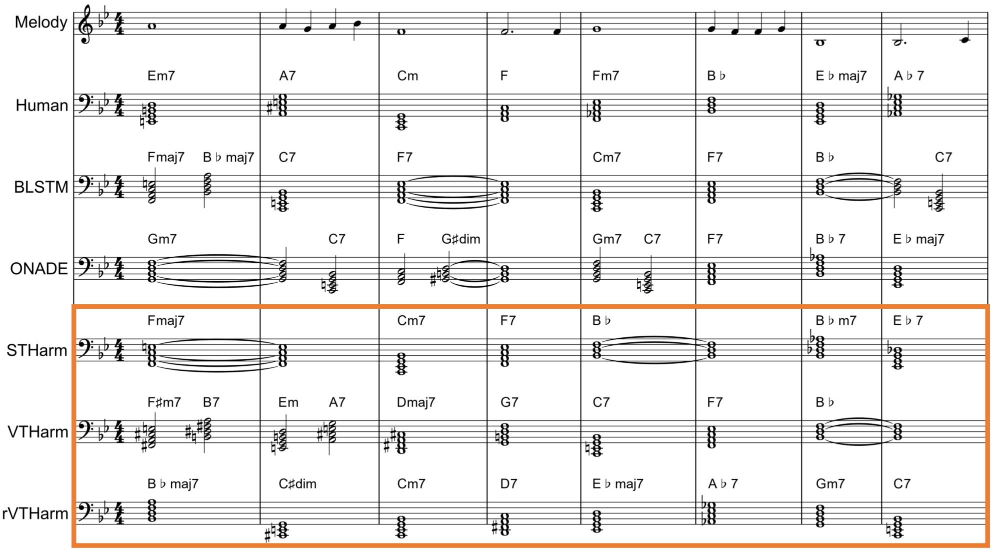


Fig. 3.4 The generated samples of the five models and the human-composed chords given the melody from the song "Stella by Starlight." The orange box emphasizes the results from the three proposed models in which the harmonic rhythms follow the binary metrical structure. In contrast, the baseline models show the syncopated rhythms for some chords.

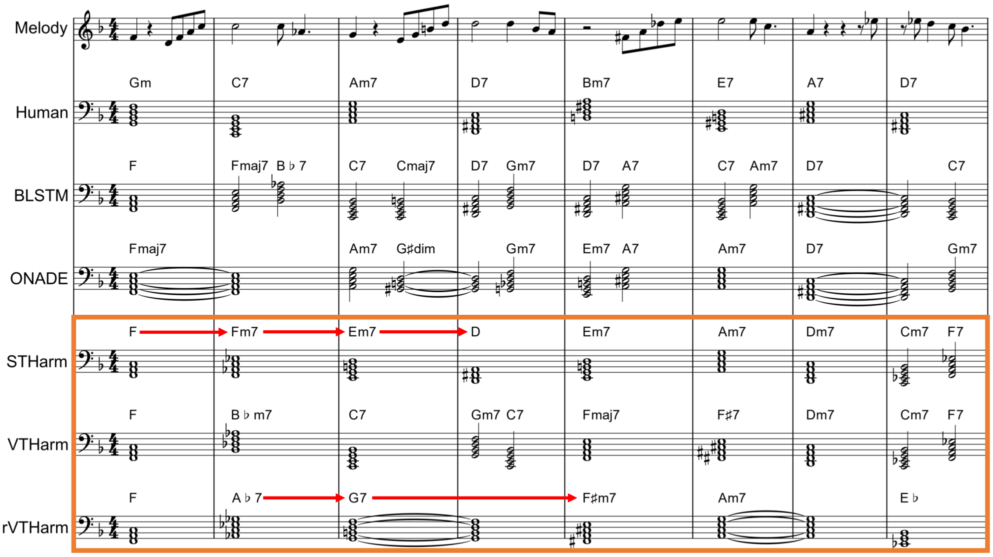


Fig. 3.5 The generated samples of the five models and the human-composed chords given the melody from the song "Shiny Stockings". The orange box focuses on the results from the three proposed models in which the chord roots progress along the circle-of-fifths rule. The red arrows indicate the chromatic progressions where the chord notes descend or ascend by intervals of a major or minor second. These progressions are related to the given melody, where a certain pattern also develops chromatically.

models show some syncopated rhythms, which can weaken the metrical boundaries. Fig. 3.5 illustrates another advantage of the proposed models, which is that the majority of the chord roots tend to shift in intervals either of perfect fourth or fifth according to the circle-of-fifths rule. This aspect reflects conventional Western music theory, which serves as domain knowledge for modeling real-world music [158, 155]. Moreover, the proposed models are shown to generate some natural chromatic progressions according to the given melody. On the other hand, the baseline models show some short transitions on the circle-of-fifths at arbitrary spots, in contrast to the melody with regular phrasings.

We also take a closer look on some attention maps between the melody inputs and their corresponding chord outputs. Figs. 3.6 and 3.7 illustrate two attention matrices that are extracted from rVTHarm after it generates two sequences of chords from the latent representations regularized by $\alpha = -3$ or $\alpha = 3$, respectively. These attention matrices are summed along their heads. According to the figures, each chord is generated by attending to certain parts of the melody notes. Although some of the focused parts do not precisely represent the right chord notes for the corresponding chord labels, it can be implied that the transformer-based architecture allows the model to pick important parts of the key for the query. This is different from the previous RNN-based methods, where every entry within a unit (e.g. bar) is aggregated into a histogram and directly mapped to a chord label. Moreover, the figures reveal a visual difference induced by varying a value of α. When $\alpha = -3$, the focused parts tend to maintain the same regardless of the chord labels. When $\alpha = 3$, on the other hand, the attention varies more often by chord labels. Consequently, the transformer-based models can be beneficial for extracting the focal elements

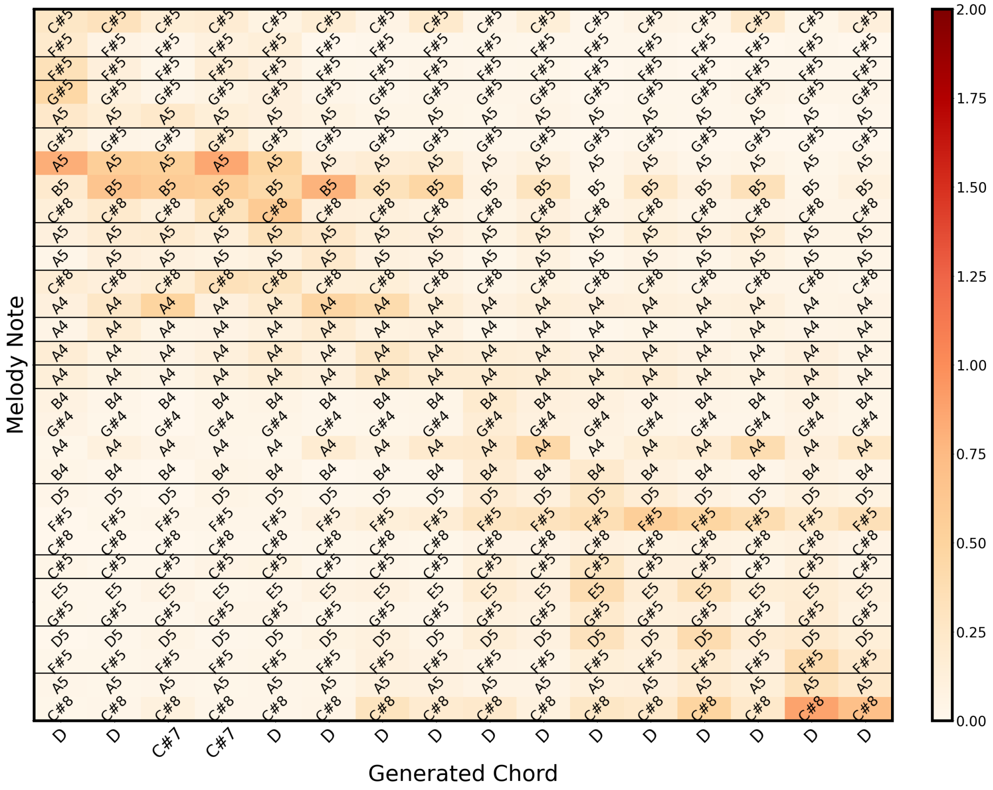


Fig. 3.6 The attention map extracted from rVTHarm where the latent representation is regularized with $\alpha = -3$. The given melody is from the song "Liza". Black vertical lines denote the boundaries of the bars, and the color denotes the values of the attention weights.

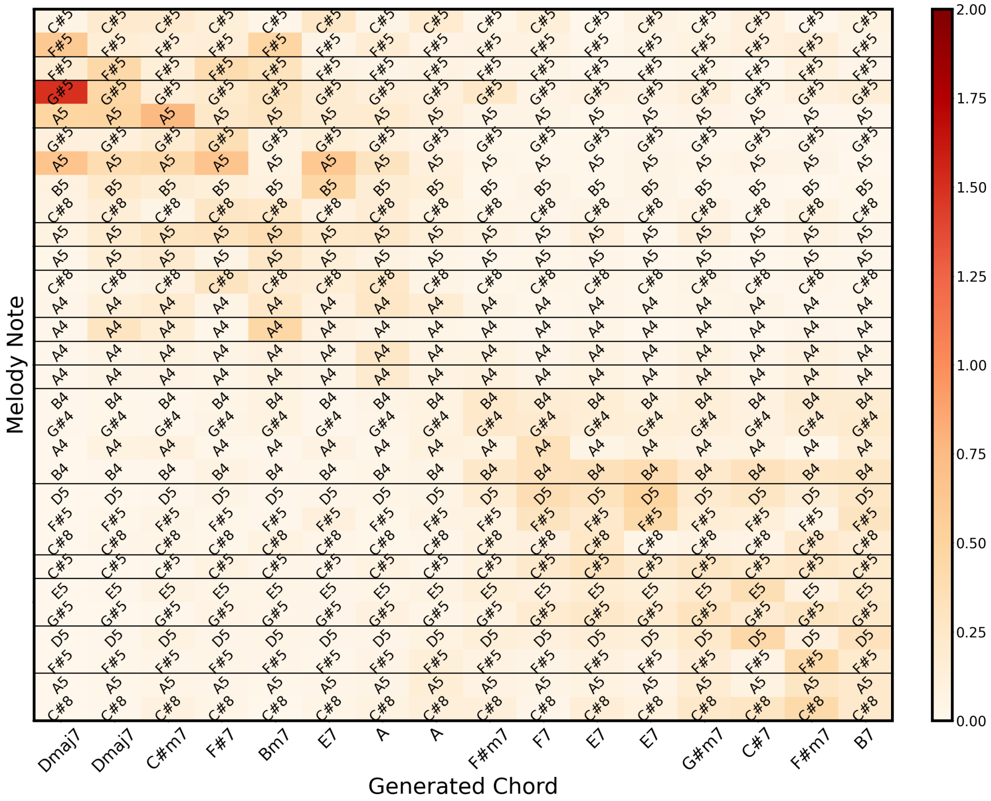


Fig. 3.7 The attention map extracted from rVTHarm where the latent representation is regularized with $\alpha = 3$. The given melody is from the song "Liza". Black vertical lines denote the boundaries of the bars, and the color denotes the values of the attention weights.

Dataset	Chord Melody Dataset		
Metric	LD↓	TPSD↓	DICD↓
VT w/o c	0.90(±0.12)	2.76(±1.02)	122.75(±36.91)
VT w/ c	**0.86(±0.14)**	**2.72(±1.02)**	**121.08(±36.82)**
rVT w/o c	0.93(±0.10)	2.86(±1.32)	124.14(±35.79)
rVT w/ c	**0.86(±0.15)**	**2.71(±1.17)**	**118.01(±36.59)**
Dataset	Hooktheory Lead Sheet Dataset		
Metric	LD↓	TPSD↓	DICD↓
VT w/o c	0.79(±0.15)	**2.53(±1.13)**	100.02(±33.35)
VT w/ c	**0.77(±0.16)**	2.54(±1.15)	**98.55(±33.53)**
rVT w/o c	0.80(±0.16)	2.39(±1.21)	93.90(±34.65)
rVT w/ c	**0.79(±0.16)**	**2.32(±1.26)**	**91.63(±34.92)**

Table 3.8 Evaluation results of the chord similarity metrics according to adding the condition token c. VT and rVT denote VTHarm and rVTHarm, respectively.

of the melody to predict its harmonic structure, different from the RNN-based approaches, and the focused elements can vary according to how fine or coarse the harmonic structure is.

3.4.6 Ablation Study

We conduct an ablation study to verify the benefit of adding the conditional token c to VTHarm and rVTHarm. We assume that c provides key signature information that can efficiently constrain the latent space to a concrete harmonic context, improving the chord structuredness and reconstruction performance of the model. We compute the chord similarity metrics between the ground truth and generated chords from the VT models according to the presence of c. The results are demonstrated in Table 3.8. This table shows that the VT models without c mostly obtain worse scores for all similarity metrics than the models with c. This indicates that adding key signature information to the VT models in most cases not only enhances the one-by-one accuracy but also improves the structure of the generated chords to be more human-like.

3.5 Conclusion and Future Work

In this paper, we have proposed melody harmonization models using the standard Transformer (STHarm), variational Transformer (VTHarm), and regularized variational Transformer (rVTHarm). We show that STHarm can create structured chords that are more human-like than LSTM-based models. VTHarm and rVTHarm can also generate more plausible chords than the baseline models with the comparable chord diversity, especially when the melody is familiar. Furthermore, rVTHarm can control chord outputs with the disentangled representation for the intended attribute. These transformer-based models can also effectively provide insights on which part of melody is important for the harmonic context, with their attention mechanism. This can further be developed into a tool for extracting a harmonic skeleton of a musical passage which can correspond to the music summarization methods. One of the common methodologies for this objective is the Schenkerian analysis which parses a musical structure according to chord grammar and chord significance [160]. Hence, they can be more beneficial than the previous RNN models in that the proposed models can be more interpretable in terms of music theories. Moreover, the proposed models allow conditional inputs, which currently are the key-signature information or latent representation, at the beginning of the decoder input. Hence, the objectives of the proposed models can be expanded to melody harmonization conditioned by a composer or genre by allowing to add prefixes representing various meta information.

Our study has several limitations that need to be investigated in the future. First, STHarm and VTHarm are not verified on the controllability of the representation which is the second task of interest in this thesis. Specifically, STHarm may only satisfy our first aim of improving generation quality from the previous approaches. Further study of the controllability of these models is necessary to fully verify the transformer-based architecture: we need to examine whether the representations can reflect aspects related to the musical attributes, even without direct regularizations. Also, our study is still limited to the shallow investigation of the connection between controllable attributes and melody awareness. Therefore, we plan to deeply explore the effect of melody awareness for more persuasive melody harmonization. Additionally, a definition of "chord complexity" is naive within this study. Thus, we need to deeply consider which aspect of music can derive harmonic complexity and how we should extract the corresponding attribute from the chord sequence [161]. The chord complexity itself should also be verified in terms of its suitability as a controllable element, as implicated by the listening test. The controllable element should fundamentally satisfy any user who wants to get plausible chords with the given melody. Furthermore, the dataset used in this study is limited to contemporary genres, which are either jazz or pop. We can expand the dataset to Western Classical music, to not only help the music makers but also to deeply examine the role of the proposed models as the analysis tools for musicologic studies. We can further compare the models with the Schenkerian analysis methods.

Chapter 4

Sketching the Expression: Flexible Rendering of Expressive Piano Performance with Self-supervised Learning

4.1 Introduction

Computational modeling of expressive music performance focuses on mimicking human behaviors that convey the music [3, 41]. For piano performance, one common task is to *render* an expressive performance from a quantized musical score. It aims to reproduce the loudness and timing of musical notes that fits to the given score. Most of the conventional studies have used musical scores of Western piano music that includes sufficient amount of guidelines for musical expressions [46, 97, 16, 42]. Recent studies using deep learning methods have successfully rendered plausible piano performances that are comparable to those of professional pianists from the given Classical scores [48, 18, 100].

More recently, it has increased attention to *controlling* music performance by manipulating one or more *disentangled* representations from a generative model. These representations are sensitive to the variation of certain factors while invariant to other factors [63]. Maezawa *et al.* aimed to control a performer's interpretation through a conditional variational recurrent neural network (CVRNN) [57]. They intended to disentangle a time-variant representation of the personal interpretation. In the acoustic domain, Tan *et al.* proposed a generative model based on a Gaussian mixture variational autoencoder (GM-VAE) that separately controlled dynamics and articulations of the notes [137]. Their novelty lied in learning multiple representations of high-level attributes from the low-level spectrogram.

However, these studies have constrained musical creativity. Maezawa *et al.* controlled musical expression only through quantized features from the musical scores. Tan *et al.* did not consider controlling tempo or timing with a latent representation. These methods may have restricted any potential for rendering piano performances with flexible musical expression. Musical creativity can be expanded not only by composers but also by performers who can elastically choose various strategies to highlight multiple nuances or emotions [162, 163, 164]. Moreover, the music generation field can be also broadened if static music created by automatic composition systems can be easily colored with realistic and elastic expression [7].

Therefore, we attempt a new approach that renders piano performances with flexible musical expressions. We disregard a typical assumption from previous studies that a performer must follow a composer's intent [2, 92, 54, 97]. According to the literature, performers learn to identify or imitate "expressive models", or *explicit planning*, of existing piano performances [165]. We focus on this attribute, defining it as a higher-level *sketch* of the expressive attributes (i.e. dynamics, articulation, and tempo [166]) that the performer draws based on a personal interpretation of the musical piece [165, 97, 57]. We also assume that the remaining attribute represents common performing strategies that are connected to certain musical patterns, while these strategies slightly differ across performers [167, 168]. We call this attribute as a *structural attribute* that belongs to given note structures of a musical piece.

In this study, we propose a generative model that can flexibly control the entire musical expression, or the explicit planning, of symbolic piano performance. Our system is based on a conditional variational autoencoder (CVAE) that is modified for sequential data [80, 57]. The system generates multiple parameters of piano performance from a note structure of a musical passage, using disentangled representations for the explicit planning and structural attribute. The source code of our system is available at `https://github.com/rsy1026/sketching_piano_expression`.

We employ a self-supervised learning framework to force the latent representations to learn our target attributes [64, 124, 80]. In addition, we facilitate independent control of the three expressive attributes–dynamics, articulation, and tempo–by utilizing an existing method that aligns the latent code with a target attribute [81, 60]. Finally, we design a novel mechanism that intuitively models a polyphonic structure of piano performance. In particular, we insert intermediate steps for *chordwise* encoding and decoding of the piano performance to our encoder-decoder architecture, where a *chord* denotes a group of simultaneous notes.

Our approach has several contributions as follows: 1) Our system aims to control musical expression while maintaining any characteristics induced by a given musical structure; 2) We use self-supervised learning where new supervisory signals are involved in regularizing the latent representations effectively; 3) Our system aims to control multiple expressive attributes independently of each other; 4) Lastly, we leverage an intermediate step that projects a notewise representation into the chordwise in the middle of our system to intuitively model the polyphonic structure of piano performance.

4.2 Proposed Methods

We aim to build a generative model that factorizes expressive piano performance as the explicit planning and structural attribute. The model is based on a conditional variational autoencoder (CVAE) that reproduces performance parameters based on a given musical structure.

4.2.1 Data Representation

We extract features that represent a human performance and the corresponding musical score, following the conventional studies [54, 169, 57].

Performance Features. We extract three features that represent the expressive attributes of each performed note, respectively: **MIDIVelocity** is a MIDI velocity value that ranges from 24 to 104. **IOIRatio** represents an instantaneous variation in tempo. We compute an inter-onset-interval (IOI) between the onset of a note and the mean onset of the *previous* chord for both a performed note and the corresponding score note. Then, a ratio of performed IOI to score IOI is calculated, clipped between 0.125 and 8, and converted into a logarithmic scale [97]. **Articulation** represents how much a note is shortened or lengthened compared to the instantaneous tempo. It is a ratio of a performed duration to an IOI value between the onset of a note and mean onset of the *next* chord [54]. It is clipped between 0.25 and 4 and converted into a logarithmic scale.

Score Features. The features for a musical score represent eight categorical attributes for how the notes are composed: **Pitch** is a MIDI index number that ranges from 21 to 108. **RelDuration** and **RelIOI** are 11-class attributes of a quantized duration and IOI between a note onset and a previous chord, respectively. They range from 1 to 11, and each class represents a multiple of a 16th note's length with respect to a given tempo [170, 15]. **IsTopVoice** is a binary attribute of whether the note is the uppermost voice. It is heuristically computed regarding pitches and durations of surrounding notes. **PositionInChord** and **NumInChord** are 11-class attributes of a positional index of a note within its chord and the total number of notes in that chord, respectively, that range from 1 to 11. An index 1 for PositionInChord denotes the most bottom position. **Staff** is a binary attribute of the staff of a note, either of the G clef or F clef. **IsDownbeat** is a binary attribute of whether a note is at a downbeat or not.

4.2.2 Modeling Musical Hierarchy

Inspired by previous studies [97, 100, 18, 69], we build a two-step encoder and decoder: An encoder models both notewise and chordwise dependencies of the inputs, and a decoder reconstructs the notewise dependency from the chordwise representation and the notewise condition. We denote a *chord* as a group of notes that are hit simultaneously, regardless of the staff, so that they sound together at an instant time [38]. Thus, learning the chordwise dependency is analogous to direct modeling of the temporal progression of the piano performance. Let $\mathcal{M} \in \mathbb{R}^{C \times N}$ be a matrix that aligns serialized notes to their polyphonic structure, where C and N are the number of chords and the number of notes, respectively. Within the encoder, the notewise representation is sequentially average-pooled by $\mathcal{M}$ with dynamic kernel sizes where each size represents the number of notes in each chord. We denote this operation as *N2C*. In this way, we can directly model chord-level dependency of the note-level expressive parameters [69]. In contrast, the decoder extends the chordwise representation from the encoder back to the notewise using the transposed alignment matrix $\mathcal{M}^T$, of which process we denote as *C2N*. Along this, the notewise embedding of the score features replenishes the notewise information for the output. Consequently, notes in the same chord *share* any information of their corresponding chord, while maintaining their differences by the conditional score features:

$$\mathrm{N2C}(e) = \frac{\mathcal{M} \cdot e}{\sum_{n=1}^{N} \mathcal{M}_{n,1:C}}, \quad \mathrm{C2N}(e) = \mathcal{M}^{\mathrm{T}} \cdot e \tag{4.1}$$

where e denotes a notewise or chordwise representation.

4.2.3 Overall Network Architecture

Our proposed network is generally based on the conditional VAE framework [142, 149]. Concretely, we use the sequential VAE that is modified for generation of sequential data [61, 80, 57]. Let $x = \{x_n\}_{n=1}^{N}$ be a sequence of the performance features, and $y = \{y_n\}_{n=1}^{N}$ be a sequence of the conditional score features. Our network has two *chordwise* latent variables $z^{(\mathrm{pln})} = \{z_c^{(\mathrm{pln})}\}_{c=1}^{C} \in \mathbb{R}^{C \times d^{(\mathrm{pln})}}$ and $z^{(\mathrm{str})} = \{z_c^{(\mathrm{str})}\}_{c=1}^{C} \in \mathbb{R}^{C \times d^{(\mathrm{str})}}$ that represent explicit planning and structural

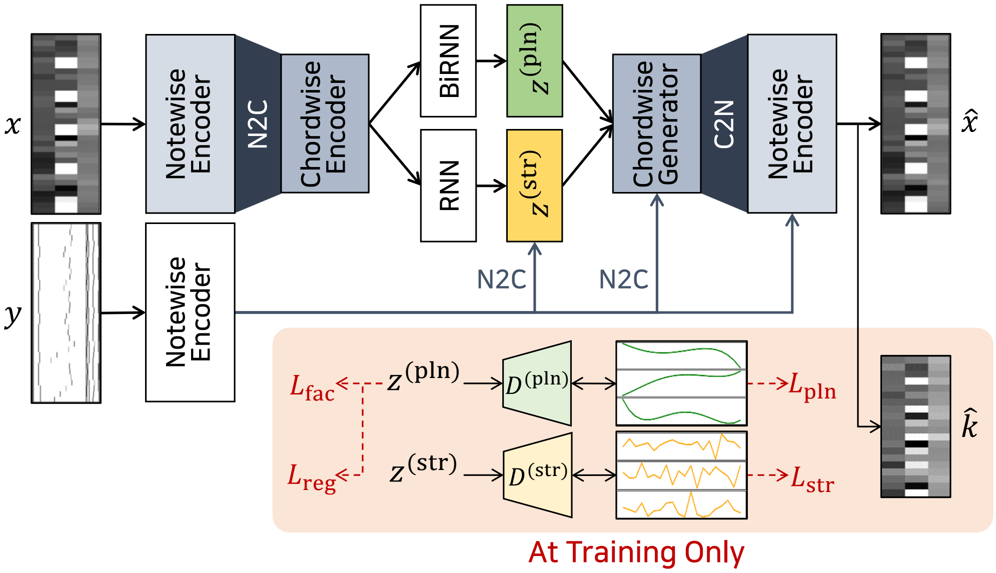


Fig. 4.1 Overall architecture of the proposed system. The orange box includes the auxiliary tasks only for training.

attribute, where $d^{(\text{pln})}$ and $d^{(\text{str})}$ are the sizes of $z^{(\text{pln})}$ and $z^{(\text{str})}$, respectively. Our network generates notewise performance parameters x from these latent variables and given score features y. The overall architecture of our proposed system is illustrated in Fig. 4.1.

Generation

A probabilistic generator parameterized by θ produces the note-level performance parameters x from the two latent variables $z^{(\text{pln})}$ and $z^{(\text{str})}$ with the given condition y. We note that the latent variables are in chord-level. This decreases a computational cost and also enables intuitive modeling of polyphonic piano performance where each time step represents a stack of notes and the simultaneous notes share common characteristics [18]:

$$\begin{aligned} p_\theta(x, y, z^{(\text{pln})}, z^{(\text{str})}) =& p_\theta(x|z^{(\text{pln})}, z^{(\text{str})}, y) \\ & p_\theta(z^{(\text{pln})}) \prod_{c=1}^{C} p_\theta(z_c^{(\text{str})}|z_{<c}^{(\text{str})}, y_{\leq c}^{(\text{chd})}) \end{aligned} \tag{4.2}$$

where $y^{(\text{chd})} = \text{N2C}(e_y)$ is the chordwise embedding, and e_y is the notewise embedding for y. We assume that the prior of $z_c^{(\text{pln})}$ is a standard normal distribution. In contrast, $z_c^{(\text{str})}$ is sampled from a sequential prior [171, 61, 80], conditioned on both previous latent variables and chordwise score features: $z_c^{(\text{str})} \sim \mathcal{N}(\mu^{(\text{prior})}, \text{diag}(\sigma^{(\text{prior})^2})$, where $[\mu^{(\text{prior})}, \sigma^{(\text{prior})}] = f^{(\text{prior})}(z_{<c}^{(\text{str})}, y_{\leq c}^{(\text{chd})})$, and $f^{(\text{prior})}$ is a unidirectional recurrent neural network. The latent representations and $y^{(\text{chd})}$ pass through the decoder as shown in Fig. 4.1. During training, the model predicts the intermediate chordwise output that is computed as N2C(x). This is to enhance reconstruction power of our system, propagating accurate information of chord-level attributes to the final decoder. The intermediate activation is then extended to the notewise through the C2N operation. The note-level parameters are generated autoregressively based on this activation and the notewise score feature. We use teacher forcing during training [172].

Inference

A probabilistic encoder parameterized by ϕ approximates the posterior distibutions of the latent representations $z^{(\text{pln})}$ and $z^{(\text{str})}$ from the performance input x and conditional score input y:

$$\begin{aligned} q_\phi(z^{(\text{pln})}, z^{(\text{str})}|x, y) =& q_\phi(z^{(\text{pln})}|x^{(\text{chd})}) \\ & \prod_{c=1}^{C} q_\phi(z_c^{(\text{str})}|x_{\leq c}^{(\text{chd})}, y_{\leq c}^{(\text{chd})}) \end{aligned} \tag{4.3}$$

where $x^{(\mathrm{chd})} = \mathrm{N2C}(e_x)$ is the chordwise embedding, and e_x is the notewise embedding for x. The posterior distributions of $z_c^{(\mathrm{pln})}$ and $z_c^{(\mathrm{str})}$ are approximated by distribution parameters encoded by $f^{(\mathrm{pln})}(x^{(\mathrm{chd})})$ and $f^{(\mathrm{str})}(x^{(\mathrm{chd})}, y^{(\mathrm{chd})})$, where $f^{(\mathrm{pln})}$ and $f^{(\mathrm{str})}$ are bidirectional and unidirectional recurrent neural networks, respectively. We note that $z^{(\mathrm{pln})}$ is independent of the score features y. This allows a flexible transfer of the explicit planning among other musical pieces. On the other hand, $z^{(\mathrm{str})}$ is constrained by y since the structural attributes are dependent on the note structure.

Training

We train the models p_θ and q_ϕ by approximating marginal distributions of the performance features x conditioned on the score features y. This requires to maximize negative evidence lower bound (ELBO) that includes regularization force by Kullback–Leibler divergence [142]:

$$\begin{aligned}
\mathcal{L}_{\mathrm{VAE}} &= \mathbb{E}_{q_\phi(z^{(\mathrm{pln})}, z^{(\mathrm{str})}|x,y)} \left[\log p_\theta(x|z^{(\mathrm{pln})}, z^{(\mathrm{str})}, y)\right] \\
&+ \mathbb{E}_{q_\phi(z^{(\mathrm{pln})}, z^{(\mathrm{str})}|x,y)} \left[\log p_\theta(k|z^{(\mathrm{pln})}, z^{(\mathrm{str})}, y)\right] \\
&- \mathrm{KL}(q_\phi(z^{(\mathrm{pln})}|x) \| p_\theta(z^{(\mathrm{pln})})) \\
&- \sum_{c=1}^{C} \mathrm{KL}(q_\phi(z_c^{(\mathrm{str})}|x_{\leq c}^{(\mathrm{chd})}, y_{\leq c}^{(\mathrm{chd})}) \| p_\theta(z_c^{(\mathrm{str})}|z_{<c}^{(\mathrm{str})}, y_{\leq c}^{(\mathrm{chd})}))
\end{aligned} \tag{4.4}$$

where $k = \mathrm{N2C}(x)$ is the chordwise performance features.

4.2.4 Regularizing the Latent Variables

We enhance disentanglement of the latent representations $z^{(\mathrm{pln})}$ and $z^{(\mathrm{str})}$ using four regularization tasks [80].

Prediction Tasks

We extract new supervisory signals for additional prediction tasks from the input data [80]. We define a signal of explicit planning $I^{(\text{pln})}$ as a set of smoothed contours of the expressive parameters. It is extracted as a polynomial function predicted from the chordwise performance parameters k. We also derive a signal of structural attribute as $I^{(\text{str})} = \text{sign}(k - I^{(\text{pln})})$ which represents normalized directions of the performance parameters. We train two discriminators $D^{(\text{pln})}$ and $D^{(\text{str})}$ that directly receive $z^{(\text{pln})}$ and $z^{(\text{str})}$, respectively. $D^{(\text{pln})}$ is composed of A sub-discriminators where each discriminator $D_a^{(\text{pln})}$ predicts a signal $I_a^{(\text{pln})}$ for each expressive attribute a from $z_a^{(\text{pln})} \in \mathbb{R}^{C \times (d^{(\text{pln})}/A)}$, where $z_a^{(\text{pln})}$ is a constituent part of $z^{(\text{pln})}$, and A is the number of expressive attributes. This setting is for a clear disentanglement among the expressive attributes. On the other hand, $D^{(\text{str})}$ predicts the signal $I^{(\text{str})}$ at once for all expressive attributes that belong to the same musical structure. All discriminators are jointly trained with the generative model, and the costs $\mathcal{L}_{\text{pln}}$ and $\mathcal{L}_{\text{str}}$ are minimized as $\mathcal{L}_{\text{pln}} = \frac{1}{A}\sum_a \text{MSE}(D_a^{(\text{pln})}(z_a^{(\text{pln})}), I_a^{(\text{pln})})$ and $\mathcal{L}_{\text{str}} = \text{MSE}(D^{(\text{str})}(z^{(\text{str})}), I^{(\text{str})})$, respectively.

Factorizing Latent Variables

We further constrain a generator to guarantee that $z^{(\text{pln})}$ delivers correct information regardless of $z^{(\text{str})}$ [119]. During training, we sample a new output $\tilde{x}$ using $z^{(\text{pln})} \sim q_\phi(z^{(\text{pln})}|x)$ and $\tilde{z}^{(\text{str})} \sim p_\theta(z^{(\text{str})})$. Then, we re-infer $\tilde{z}^{(\text{pln})} \sim q_\phi(\tilde{z}^{(\text{pln})}|\tilde{x})$ to estimate the superversory signal $I^{(\text{pln})}$. This prediction loss is backpropagated only through the generator:

$$\mathcal{L}_{\text{fac}} = \frac{1}{A}\sum_a \text{MSE}(D_a^{(\text{pln})}(\tilde{z}_a^{(\text{pln})}), I_a^{(\text{pln})}) \tag{4.5}$$

Aligning Latent Variables with Factors

Finally, we enable the "sliding-fader" control of the expressive attributes [60]. To this end, we employ the regularization loss proposed by Pati *et al.* [81] that aligns specific dimensions of $z^{(\mathrm{pln})}$ with the target expressive attributes. This method assumes that a latent representation can be disentangled through its monotonic relationship with a target attribute. Let d_i and d_j be a target dimension d of ith and jth latent representations, respectively, where $d \in z_a^{(\mathrm{pln})}$, $i, j \in [1, M]$, and M is the size of a mini-batch. A distance matrix $\mathcal{D}_d$ is computed between d_i and d_j within a mini-batch, where $\mathcal{D}_d = d_i - d_j$. A similar distance matrix $\mathcal{D}_a$ is computed for the two target attribute values a_i and a_j. We minimize a MSE between $\mathcal{D}_d$ and $\mathcal{D}_a$ as follows:

$$\mathcal{L}_{\mathrm{reg}} = \mathrm{MSE}(\tanh(\mathcal{D}_d), \mathrm{sign}(\mathcal{D}_a)) \tag{4.6}$$

4.2.5 Overall Objective

The overall objective of our proposed network aims to generate realistic performance features with properly disentangled representations for the intended factors:

$$\mathcal{L} = \mathcal{L}_{\mathrm{VAE}} + \lambda_{\mathrm{pln}}\mathcal{L}_{\mathrm{pln}} + \lambda_{\mathrm{str}}\mathcal{L}_{\mathrm{str}} + \lambda_{\mathrm{fac}}\mathcal{L}_{\mathrm{fac}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}} \tag{4.7}$$

where λ_{pln}, λ_{str}, λ_{fac}, and λ_{reg} are hyperparameters for balancing the importance of the loss terms.

4.3 Experimental Settings

4.3.1 Dataset and Implementation

We use Yamaha e-Competition Dataset [18] and Vienna 4x22 Piano Corpus [173]. From these datasets, we collect 356 performances of 34 pieces by Frédéric Chopin, which have been representative research subjects for analyzing the Western musical expression [167, 174, 42, 17]. We use 30 pieces (108,738 batches) for training and the rest for testing. To verify the generality of model performances, we also collect the external dataset from ASAP dataset [175]. We use 116 performances for 23 pieces by 10 composers who represent various eras of Western music. For subjective evaluation, we collect 42 songs of non-Classical songs from online source[1] which are less constrained to written expression than most Classical excerpts.

We basically follow Jeong *et al.* [18] to compute the input features from the aligned pairs of performance and score data. We set MIDI velocities and Beat Per Minute (BPM) of all notes in the score data to be 64 and 120, respectively. We also remove any grace notes for simplicity and manually correct any errors. The performance features are further normalized into a range from -1 to 1 for training. We compose each batch by slicing an entire piece into short excerpts where notes for maximum 16 chords are contained and 12 chords overlap. Thus, a length of each batch varies from 16 to 114. We set a degree of the polynomial function computing $I^{(\mathrm{pln})}$ as 4 through an ablation study.

The embedding sizes of the performance input x and score input y are 256 and 128, respectively. The sizes of $z^{(\mathrm{pln})}$, $z^{(\mathrm{str})}$, and hidden dimension are 12 and 64, and 256, respectively. We use an ADAM optimizer [151] with an initial

[1] http://www.ambrosepianotabs.com/page/library

learning rate of 1e-5, which is reduced by 5% every epoch during backpropagation. We empirically set λ_{pln}, λ_{str}, λ_{fac}, and λ_{reg} to be 1000, 100, 1, 10, respectively. We train all models for 100 epochs (170,000 iterations) with a batch size of 64. For quantitative evaluation, each model repeatedly generates 20 samples for the same inputs considering the randomness of each result. For subjective and qualitative evaluations, all models generate the samples using the truncation trick with a threshold of 2 [176].

4.3.2 Comparative Methods

To the best of our knowledge, there is no existing method that does not intentionally follow the written guidelines in the musical score. Therefore, we use variants of our proposed network as comparing methods that differ in model architecture: **Notewise** denotes the proposed model without the hierarchical learning. **CVAE** denotes a variant of Notewise where $z^{(\text{pln})}$ is substituted with the supervisory signal $I^{(\text{pln})}$. We also conduct an ablation study that investigates necessity of the four loss terms.

4.4 Evaluation

We evaluate the proposed network in terms of four criteria: the generation quality, disentanglement of the latent representations, ability to control the expressive attributes, and subjective quality from human listeners. For all figures presenting the performance features, if no explanation is present, we visualize Articulation and IOIRatio in the linear scales for convenient comparison among the features.

Dataset	Internal			External		
Metric	$\mathrm{R}_{\mathrm{recon}}$	$\mathrm{R}_{x\|\mathrm{pln}}$	$\mathrm{R}_{x\|\mathrm{pln}_0}$	$\mathrm{R}_{\mathrm{recon}}$	$\mathrm{R}_{x\|\mathrm{pln}}$	$\mathrm{R}_{x\|\mathrm{pln}_0}$
Notewise	**0.870**	0.392	0.203	**0.875**	0.479	0.177
CVAE	0.730	0.338	0.223	0.741	0.399	0.216
$\mathcal{L}_{\mathrm{pln}}$	0.627	0.357	0.229	0.687	0.414	**0.220**
$\mathcal{L}_{\mathrm{pln}} + \mathcal{L}_{\mathrm{str}}$	0.770	0.325	0.181	0.837	0.398	0.195
w/o $\mathcal{L}_{\mathrm{fac}}$	0.774	0.289	0.176	0.838	0.354	0.173
w/o $\mathcal{L}_{\mathrm{reg}}$	0.737	**0.437**	0.224	0.793	**0.502**	0.216
Ours	0.737	0.427	**0.231**	0.789	0.498	0.203

Table 4.1 Evaluation results for the generation quality. The higher score is the better.

4.4.1 Generation Quality

We compute Pearson's correlation coefficients between the reconstructed or generated samples and human piano performances [54, 42, 100, 57]. We first measure the reconstruction quality of the test samples ("$\mathrm{R}_{\mathrm{recon}}$"). Then, we evaluate the samples generated from $\tilde{z}^{(\mathrm{str})} \sim p_\theta(z^{(\mathrm{str})})$ and either of : 1) $z^{(\mathrm{pln})} \sim q_\phi(z^{(\mathrm{pln})}|x)$ ("$\mathrm{R}_{x|\mathrm{pln}}$") and 2) $z_0^{(\mathrm{pln})} \sim q_\phi(z_0^{(\mathrm{pln})}|x_0)$ ("$\mathrm{R}_{x|\mathrm{pln}_0}$"), where x_0 is a zero matrix.

The results are shown in Table 4.1. Notewise shows the best scores in both datasets, and our method outperforms CVAE in $\mathrm{R}_{\mathrm{recon}}$. It indicates that our proposed architecture where a latent representation is used instead of a direct condition is generally good at reconstructing the human data. When using the randomly sampled $\tilde{z}^{(\mathrm{str})}$, our method and the model without $\mathcal{L}_{\mathrm{reg}}$ show stable scores compared to other baseline models. The model without $\mathcal{L}_{\mathrm{reg}}$ also shows the highest scores in $\mathrm{R}_{x|\mathrm{pln}}$ for both datasets. It indicates that $\mathcal{L}_{\mathrm{reg}}$ may contribute the least to generation power among other loss terms. CVAE and the model only with $\mathcal{L}^{(\mathrm{pln})}$ also show high scores in $\mathrm{R}_{x|\mathrm{pln}_0}$. This may be due to the posterior collapse that makes the decoder depends mostly on the score

Dataset	Internal		External	
Metric	MSE_p	MSE_s	MSE_p	MSE_s
Notewise	0.003	0.006	0.022	0.028
CVAE	0.034	0.045	0.085	0.092
$\mathcal{L}_\text{pln}$	0.028	0.036	0.074	0.077
$\mathcal{L}_\text{pln} + \mathcal{L}_\text{str}$	0.012	0.015	0.022	0.027
w/o $\mathcal{L}_\text{fac}$	0.018	0.023	0.021	0.025
w/o $\mathcal{L}_\text{reg}$	0.002	0.004	0.014	0.022
Ours	**0.001**	**0.002**	**0.012**	**0.020**

Table 4.2 Evaluation results for the disentanglement of the latent representations.

condition [107], which is demonstrated in the supplementary material.

4.4.2 Disentangling Latent Representations

We verify whether the latent representations are well-disentangled by appropriate information[80]. To this end, each model infers the latent representations $z^{(\text{pln})}$ and $z^{(\text{str})}$ from the test sets. Each model also randomly samples $\tilde{z}^{(\text{str})}$ and infers $z_0^{(\text{pln})} \sim q_\phi(z^{(\text{pln})}|x_0)$. We use $z_0^{(\text{pln})}$ to measure the structural attribute, since $z_0^{(\text{pln})}$ represents a flat expression where the structural attribute can be solely exposed. Each model generates new outputs as $x^{(\text{pln})} \sim p_\theta(x^{(\text{pln})}|z^{(\text{pln})}, \tilde{z}^{(\text{str})}, y)$ and $x^{(\text{str})} \sim p_\theta(x^{(\text{str})}|z_0^{(\text{pln})}, z^{(\text{str})}, y)$. Then, we compute a new signal $\tilde{I}^{(\text{pln})}$ from $x^{(\text{pln})}$ using the polynomial regression. The MSE values are calculated as $\text{MSE}_\text{p} = \text{MSE}(\tilde{I}^{(\text{pln})}, I^{(\text{pln})})$ and $\text{MSE}_\text{s} = \text{MSE}(x^{(\text{str})}, k - I^{(\text{pln})})$.

Table 4.2 shows that our method achieves the best scores in all metrics for both datasets. This confirms that our proposed system can learn the latent representations that reflect the intended attributes. Notewise and the model without $\mathcal{L}_\text{reg}$ also show the robust scores compared to other baseline models. It indicates that using the notewise modeling alone is still relevant for achieving appropriate representations. It also implies that $\mathcal{L}_{\text{reg}}$ may not contribute to the disentanglement as much as other loss terms.

4.4.3 Controllability of Expressive Attributes

We sample a new input $\bar{x}$ where entries of each feature are constant across time. Then, each model infers $\bar{z}^{(\text{pln})} \sim q_\phi(\bar{z}^{(\text{pln})}|\bar{x})$. We control each attribute by varying dimension values of $\bar{z}^{(\text{pln})}$ following Tan *et al.* [60] and examine the new samples generated from $\bar{z}^{(\text{pln})}$. We leverage the existing metrics to measure the controllability of each model [60]: *Consistency* ("C") measures consistency across samples in terms of their controlled attributes; *restrictiveness* ("R") measures how much the uncontrolled attributes maintain their flatness over time; and *linearity* ("L") measures how much the controlled attributes are correlated with the corresponding latent dimensions.

For controlling each attribute, each model first infers $z^{(\text{pln})}$ for all test samples. Then, we compute the maximum and minimum values of the target dimension $d^{(\text{attr})} = \{d_t^{(\text{attr})}\}_{t=1}^{T}$, where $T \in \{N, C\}$. Then, we set a dimension value of each timestep as $d_t^{(\text{attr})} = \min(d^{(\text{attr})}) + \frac{t}{T}(\max(d^{(\text{attr})}) - \min(d^{(\text{attr})}))$. If an expressive attribute is controlled by $z^{(\text{pln})}$ in the appropriate way, the corresponding attribute should only change along with the target dimension values while the other attributes maintain their status. In the case of controlling dynamics of the samples, the metrics are computed as follows [60]:

$$\text{Consistency} = 1 - \frac{1}{T}\sum_{t=1}^{T} \underset{t}{\sigma}(v_{1\ldots M,t}) \tag{4.8}$$

Dataset	Internal			External		
Metric	C	R	L	C	R	L
Notewise	0.782	0.916	0.632	0.775	0.914	0.656
CVAE	0.798	0.812	0.620	0.773	0.802	0.649
$\mathcal{L}_{\text{pln}}$	0.693	0.852	0.323	0.694	0.834	0.324
$\mathcal{L}_{\text{pln}} + \mathcal{L}_{\text{str}}$	0.633	0.882	0.253	0.639	0.865	0.277
w/o $\mathcal{L}_{\text{fac}}$	0.831	0.846	0.789	0.832	0.831	0.847
w/o $\mathcal{L}_{\text{reg}}$	0.804	**0.955**	0.653	0.808	**0.946**	0.657
Ours	**0.942**	0.953	**0.976**	**0.944**	0.945	**0.977**

Table 4.3 Evaluation results for the controllability of the expressive attributes. C, R, and L denotes consistency, restrictiveness, and linearity, respectively. Each score is the average score for the expressive attributes.

$$\text{Restrictiveness} = 1 - \frac{1}{2M} \left(\sum_{m=1}^{M} \underset{m}{\sigma}(a_{m,1..T}) + \underset{m}{\sigma}(i_{m,1..T}) \right) \tag{4.9}$$

$$\text{Linearity} = R^2(\mathcal{S}(p_{1...M})) \tag{4.10}$$

where v, a, and i are respectively the values of *MIDIVelocity*, *Articulation*, and *IOIRatio* of the generated output, $\mathcal{S}$ is a linear regression model, $p_m = \{(d_t^{(\text{attr})}, v_{m,t}) | t \in [1, T]\}$, and M is the number of samples. We average over the three expressive attributes–dynamics, articulation, and tempo–into one score for each metric.

Table 4.3 demonstrates that our system shows the best scores in consistency and linearity in both internal and external datasets. This indicates that our proposed method can robustly control the latent representation $z^{(\text{pln})}$ in intended way. The model without $\mathcal{L}_{\text{reg}}$ outperforms our method in restrictiveness. It indicates that the uncontrolled attributes by this model are the least interfered by the controlled attribute. However, its scores on consistency and linearity are lower than ours. It confirms that $\mathcal{L}_{\text{reg}}$ promotes linear control of the target attributes.

Dataset	Internal	
Metrics	$\mathrm{KLD_p}$	$\mathrm{KLD_s}$
Notewise	1.2601(±0.3141)	0.5294(±0.0850)
CVAE	-	**0.0225(±0.0052)**
$\mathcal{L}_{\mathrm{pln}}$	0.9139(±0.1274)	0.0268(±0.0049)
$\mathcal{L}_{\mathrm{pln}} + \mathcal{L}_{\mathrm{str}}$	0.9468(±0.1210)	0.6041(±0.0441)
w/o $\mathcal{L}_{\mathrm{fac}}$	1.1374(±0.4421)	0.5127(±0.0395)
w/o $\mathcal{L}_{\mathrm{reg}}$	**0.8856(±0.1759)**	0.6597(±0.0545)
Ours	1.4338(±0.7836)	0.6298(±0.0495)

Dataset	External	
Metrics	$\mathrm{KLD_p}$	$\mathrm{KLD_s}$
Notewise	1.1671(±0.4194)	0.4559(±0.1084)
CVAE	-	**0.0188(±0.0055)**
$\mathcal{L}_{\mathrm{pln}}$	1.0053(±0.2099)	0.0278(±0.0061)
$\mathcal{L}_{\mathrm{pln}} + \mathcal{L}_{\mathrm{str}}$	1.0528(±0.2610)	0.6260(±0.0539)
w/o $\mathcal{L}_{\mathrm{fac}}$	1.4065(±0.6481)	0.5329(±0.0474)
w/o $\mathcal{L}_{\mathrm{reg}}$	**0.9773(±0.2403)**	0.6688(±0.0602)
Ours	1.7641(±0.9871)	0.6337(±0.0581)

Table 4.4 Evaluation results for the KL divergence loss.

4.4.4 KL Divergence

Table 4.4 shows the results for the KL divergence of $z^{(\mathrm{pln})}$ and $z^{(\mathrm{str})}$ which we denote as "$\mathrm{KLD_p}$" and "$\mathrm{KLD_s}$", respectively. It shows that our method reveals the highest KL divergence of both latent variables in both datasets. In contrast, the model without $\mathcal{L}_{\mathrm{reg}}$ and CVAE shows the lowest values for $\mathrm{KLD_p}$ and $\mathrm{KLD_s}$, respectively. In particular, CVAE and the model only with $\mathcal{L}_{\mathrm{pln}}$ show abrupt decreases in $\mathrm{KLD_s}$ compared to other models. It shows that these models have extremely small regularization power: it may have led to the posterior collapse of any information that $z^{(\mathrm{str})}$ should carry and allowed the decoder to become mostly dependent on the deterministic condition of the score features [107].

Dataset	Internal			External		
Metrics	$\mathrm{R_{recon}}$	$\mathrm{R_{x\|pln}}$	$\mathrm{R_{x\|pln_0}}$	$\mathrm{R_{recon}}$	$\mathrm{R_{x\|pln}}$	$\mathrm{R_{x\|pln_0}}$
$d_{I^{(\mathrm{pln})}} = 1$	0.735	0.298	0.218	0.784	0.330	**0.214**
$d_{I^{(\mathrm{pln})}} = 2$	0.717	0.348	0.225	0.769	0.394	0.204
$d_{I^{(\mathrm{pln})}} = 4$	**0.737**	0.427	**0.231**	**0.789**	0.498	0.203
$d_{I^{(\mathrm{pln})}} = 8$	0.719	**0.546**	0.197	0.786	**0.650**	0.211

Table 4.5 Evaluation results for the generation quality according to the degree of the polynomial function. The higher score is the better.

Dataset	Internal		External	
Metrics	$\mathrm{MSE_p}$	$\mathrm{MSE_s}$	$\mathrm{MSE_p}$	$\mathrm{MSE_s}$
$d_{I^{(\mathrm{pln})}} = 1$	**0.0013**	0.0150	**0.0021**	0.0229
$d_{I^{(\mathrm{pln})}} = 2$	0.0024	0.0149	0.0028	0.0243
$d_{I^{(\mathrm{pln})}} = 4$	**0.0013**	**0.0115**	**0.0021**	0.0196
$d_{I^{(\mathrm{pln})}} = 8$	0.0017	0.0127	0.0022	**0.0172**

Table 4.6 Evaluation results for the disentanglement of the latent representations according to the degree of the polynomial function.

4.4.5 Ablation Study

We also conduct an ablation study for the degree of the polynomial function to compute $I^{(\mathrm{pln})}$. We investigate the cases where the degree $d_{I^{(\mathrm{pln})}}$ is 1, 2, 4, or 8. Tables 4.5 and 4.6 show the results for the metrics of the generation quality and disentanglement of the representations, respectively. In the generation quality, $d_{I^{(\mathrm{pln})}} = 4$ receives the best scores for the three metrics out of the six, compared to other models. In particular, $d_{I^{(\mathrm{pln})}} = 4$ shows the highest reconstruction scores in both datasets, whereas $d_{I^{(\mathrm{pln})}} = 8$ shows the best scores for $R_{x|\mathrm{pln}}$ in both datasets. $d_{I^{(\mathrm{pln})}} = 4$ also shows the best score for $R_{x|\mathrm{pln}_0}$ in the internal dataset. However, $d_{I^{(\mathrm{pln})}} = 1$ is the highest for $R_{x|\mathrm{pln}_0}$ in the external dataset instead of $d_{I^{(\mathrm{pln})}} = 8$. In the disentanglement metrics, nonetheless, our method with $d_{I^{(\mathrm{pln})}} = 4$ shows the best scores for most metrics. The model with $d_{I^{(\mathrm{pln})}} = 1$ shows the best scores for $\mathrm{MSE_p}$ in both datasets but relatively low scores for

MSE_s. According to these results, we determine the degree of the polynomial function for $I^{(\text{pln})}$ to be 4 in this study.

4.4.6 Subjective Evaluation

We conduct a listening test to compare the proposed model architecture to Notewise and CVAE. We qualitatively evaluate the base quality of the samples that have flat expressions, so that quality judgments are independent of any preference of arbitrary explicit planning. We generate each sample using $z_0^{(\text{pln})}$. A listening test is composed of 30 trials where each participant chooses a more "human-like" sample out of the generated sample and its plain MIDI [100]. Both samples have the same length which is a maximum of 15 seconds, rendered with TiMidity++[2] without any pedal effect. *Human-likeness* denotes how similar the sample is to an actual piano performance that commonly appears in popular music. A total of 28 participants are involved, and 6 participants are professionally trained in music.

The results are demonstrated in Table 4.7 and Fig. 4.2. We first measure a *winning rate*, a rate of winning over the plain MIDI. Winning rate is computed for each model as a ratio of the number of winning the plain MIDI to the total number of trials per each participant. We also compute a *top-ranking rate*, a rate of being the highest rank among the three models in terms of winning rate. Top-ranking rate is calculated as a ratio of the number of participants who choose the corresponding model most frequently among the three models to the total number of participants in each group. Concretely, the number of being top-ranked by each participant is counted by $1/numModel$, where $numModel$

[2] https://sourceforge.net/projects/timidity/

Metric Group	Winning Rate (Human-likeness) T	 UT	 Overall
Notewise	0.317(±0.223)	0.541(±0.316)	0.493(±0.309)
CVAE	**0.467(±0.356)**	0.477(±0.342)	0.475(±0.338)
Ours	0.417(±0.256)	**0.555(±0.256)**	**0.525(±0.258)**

Table 4.7 Evaluation results for the winning rate in terms of human-likeness. T, UT, and Overall denote musically trained, untrained, and all groups, respectively.

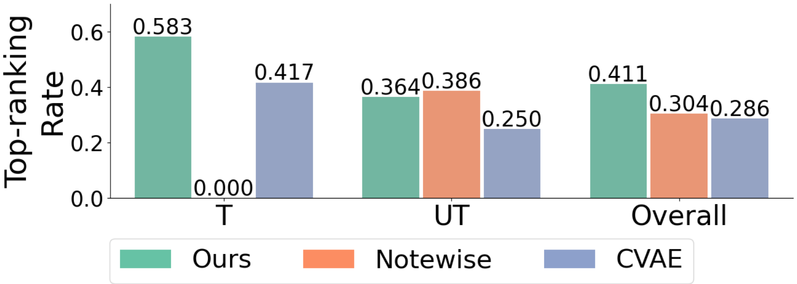


Fig. 4.2 Evaluation results for the top-ranking rate. T, UT, and Overall denote musically trained, untrained, and all groups, respectively.

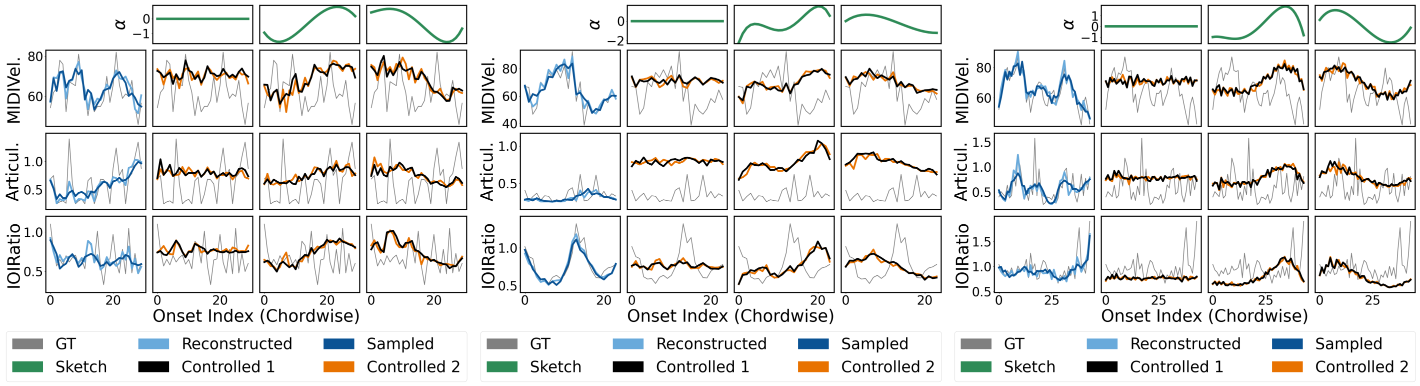
denotes the number of models that are being top-ranked at the same time.

The results show that musically *trained* ("T") and *untrained* ("UT") groups show the different tendency of each other: in the trained group, CVAE shows the best winning rate, and our method gets the best top-ranking rate; in the untrained group, our method shows the highest winning rate, whereas Notewise is top-ranked most frequently. We note that our system reveals smaller variances than those of CVAE and Notewise of the musically trained and untrained groups in the winning rate, respectively. Moreover, our system receives the highest overall scores for both metrics. It indicates that our system can be stably perceived more human-like than the plain MIDI compared to other baseline models.

4.4.7 Qualitative Examples

Our system can render new piano performances from the scratch given a musical score. It can directly generate expressive parameters from the randomly sampled $\tilde{z}^{(\text{pln})} \sim p_\theta(z^{(\text{pln})})$ and $\tilde{z}^{(\text{str})} \sim p_\theta(z^{(\text{str})})$. We note that $\tilde{z}^{(\text{pln})}$ does not have temporal dependency: each $\tilde{z}_c^{(\text{pln})}$ is sampled independently of $\tilde{z}_{c-1}^{(\text{pln})}$. Hence, we need to insert specific values $\{\alpha^{(c)}\}_{c=1}^{C}$, which we call as "smooth sketches", into the target dimensions of $z^{(\text{pln})}$ if any temporal dependency of explicit planning is necessary. Fig. 4.3 shows that the controlled parameters are greatly correlated with α, while their local characteristics follow those of the ground truth. In addition, the black and orange lines together demonstrate granular variety in the parameters induced by different $\tilde{z}^{(\text{str})}$ for the same musical structure. Moreover, Fig. 4.4 shows that our system can estimate explicit planning from arbitrary human performances, indicating that our system can derive relevant information on explicit planning from the unseen data.

Lastly, we show the results generated by interpolating between the latent representations of two performance samples, where only one expressive attribute differs from each other. A pair of the samples is created by increasing or decreasing the original attribute values of each attribute linearly by ±30%. We sample each pair for a condition that is either $\{loud, quiet\}$, $\{staccato, legato\}$, or $\{fast, slow\}$ by modifying only dynamics, articulation or tempo, respectively. Next, $\{z_a^{(\text{pln})}, z_b^{(\text{pln})}\}$ is inferred from each pair. We interpolate between $\{z_a^{(\text{pln})}, z_b^{(\text{pln})}\}$ and produce its mixture as $z_{a,b}^{(\text{pln})} = z_a^{(\text{pln})} \times 0.5 + z_b^{(\text{pln})} \times 0.5$. Fig. 4.5 shows the results that are generated from $\{z_a^{(\text{pln})}, z_{a,b}^{(\text{pln})}, z_b^{(\text{pln})}\}$ for each condition pair. All results are based on the same $\tilde{z}^{(str)}$ that is randomly sampled. They generally show that only the target attribute changes along the direction

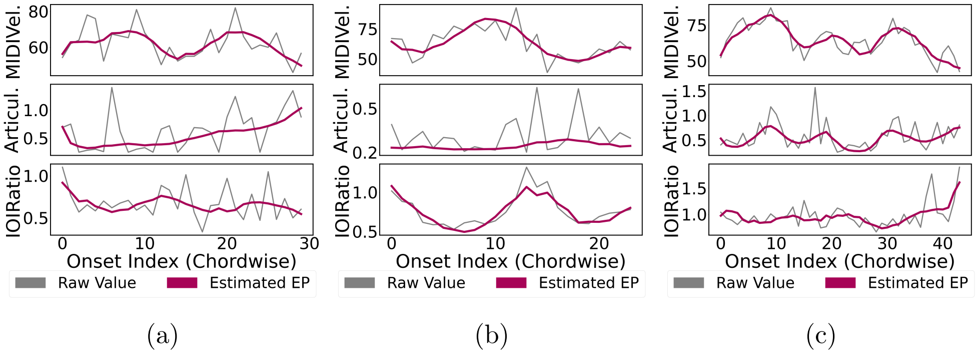


(a)

(b)

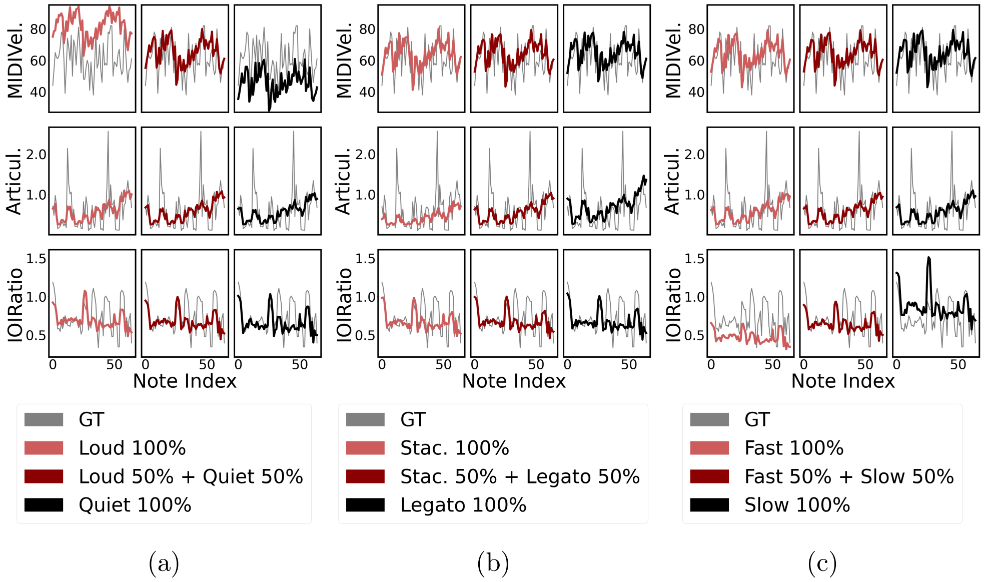
(c)

Fig. 4.3 Qualitative samples for the proposed system. MIDIVel. and Articul. are abbreviation of MIDIVelocity and Articulation from the performance features. Light-blue, blue and gray lines denote the reconstructed results, sampled results from the inferred $z^{(\mathrm{pln})}$, and their ground truths, respectively; black and orange lines denote the controlled results that are generated from different random $\tilde{z}^{(\mathrm{str})}$; and green lines denote the "sketch" values, or α, that are inserted to $z^{(\mathrm{pln})}$. The samples demonstrate three excerpts that are: (a) Haydn's Keyboard Sonata, Hob. XVI:39, 3rd movement, mm. 53-56; (b) Schubert's Impromptu, Op. 90, No. 4, mm. 149-152; and (c) Balakirev's Islamey, Op. 18, mm. 29-32.



(a) (b) (c)

Fig. 4.4 Qualitative results for estimating the explicit planning from raw piano performances. Pink and gray lines denote the estimated contours and raw performance parameters, respectively. The results in (a), (b), and (c) are from the same excerpts for (a), (b), and (c) in Fig. 4.3, respectively.



(a) (b) (c)

Fig. 4.5 Qualitative results for interpolating between the latent representations of two piano performances in paired condition. The results in (a), (b), and (c) are from the condition sets for dynamics, articulation, and tempo, respectively. Gray lines denote the original performance parameters, while pink, red, and black lines denote the parameters controlled by $z_a^{(\text{pln})}$, $z_{a,b}^{(\text{pln})}$, and $z_b^{(\text{pln})}$, respectively. The excerpt is Haydn's Keyboard Sonata, Hob. XVI:39, 3rd movement, mm. 53-56.

of interpolation, while other attributes maintain their positions. It demonstrates that $z^{(\text{pln})}$ from our model carries appropriate information with respect to the explicit planning and is well disentangled by the expressive attributes.

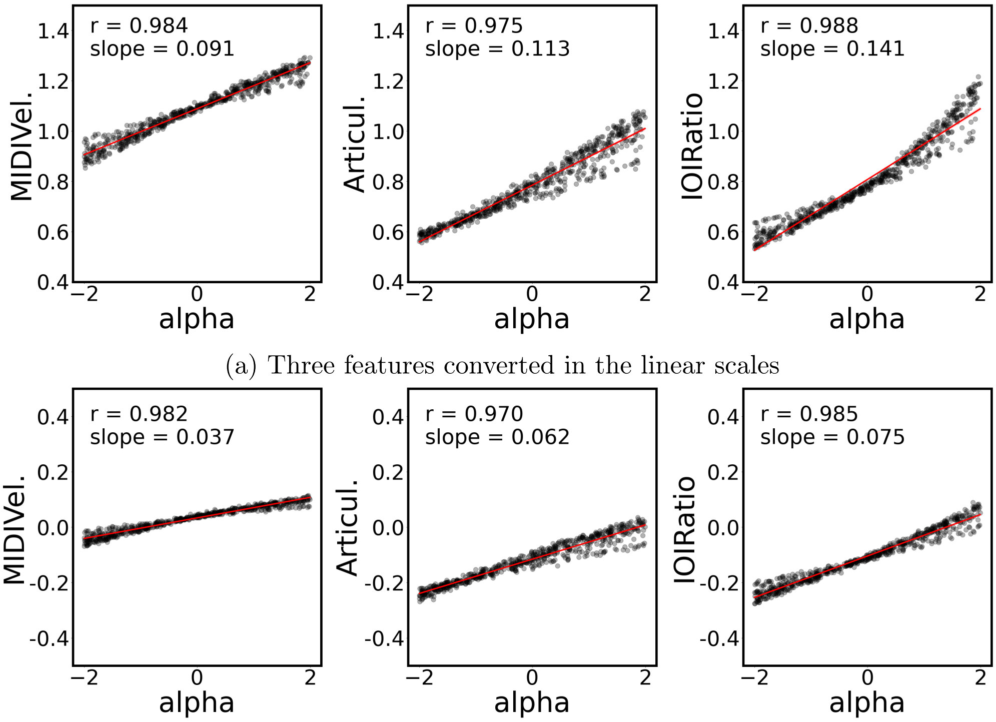
Demo and more qualitative samples are introduced in the online page [3]

4.4.8 Extent of Control

Lastly, we examine an extent of control for each expressive attribute to observe how much the target attributes can be modified differently along with the variation of α. To this end, we conduct the following procedure. First, we randomly sample $\{\tilde{z}^{(\text{pln})}$. For each latent dimension aligned with each expressive parameter, we set values as $\alpha = c$, where c is a constant sampled from the uniform distribution that ranges from [-2, 2]. We generate the performance features from this $\{\tilde{z}^{(\text{pln})}$ and take an average of the feature values over the time axis. Then, we conduct a linear regression between the averaged feature values for each attribute and the corresponding values of α. We also compute Spearman's rank correlation coefficients between them. We note that Articulations and IOIRatio are trained in logarithmic scale to regard human perception [177]. Therefore, we compare three features by collectively converting them into both linear and logarithmic scales.

Fig. 4.6 illustrates that the three expressive attributes show the different extents in the correlation between α and the resulting feature. Fig. 4.6 shows that IOIRatio shows a larger slope of the regression line compared to the other two performance features regardless of the feature scale. This may lead to more abrupt changes in tempo than other attributes even if α linearly increases or

[3] `https://free-pig-6c6.notion.site/DEMO-c20a1fea7a0844468a05b971c3b9ef3c`



(a) Three features converted in the linear scales

(b) Three features converted in the logarithmic scales

Fig. 4.6 Spearman's rank correlation coefficients (r) and slopes of the regression lines between α and values of the three performance features. Red lines denote the regression lines.

decreases. In other words, the listeners may perceive tempo variation much more than variation in other characteristics of the piano performance generated with arbitrary sketches. Balancing the extent of control during training the model in the future is necessary for more precise control of perceived musical expressions. It is particularly important when a user desires to express emotions by varying the attributes, as the effects of variations in dynamics and tempo are different in terms of conveying emotional expressiveness [178].

4.5 Conclusion

We propose a system that can render expressive piano performance with flexible control of musical expression. We attempt to achieve representations for the explicit planning and structural attribute through self-supervised learning objectives. We also leverage the two-step modeling of two hierarchical units for an intuitive generation. Experimental results confirm that our system shows stable generation quality, disentangles the target representations, and controls all expressive attributes independently of each other. Future work can be improving our system using a larger dataset for various genres and composers. We can also deeply investigate new supervisory signal I_{p} with respect to whether it can be utilized to identify a performing style of music performance. Moreover, we can further compare our system with recent piano-rendering models [18] to investigate any connections between a performer's explicit planning and a composer's intent.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The main objective of this thesis is to improve CGMC systems in terms of generation quality and controllability. The nature of music has led us to assume that generation quality can be enhanced by explicit learning of musical structure. Furthermore, the flexible control with the latent representation can help the generation systems expand musical creativity. However, the previous attempts in CGMC have not deeply investigated these two challenges: the data representations and model architectures have not been intuitive for modeling multi-dimensional music, and the former studies for controlling CGMC systems have been mostly limited to dealing with narrow ranges of controllable factors and tasks.

Therefore, on the basis of the literature backgrounds (Chapter 2), we apply several practical methods to address these challenges. For the generation quality, we provide effective ways for generative models to encode musical structure from the raw data. Concretely, we directly utilize alignment paths between the data representations and the target units for which we want to model the explicit structure. For the controllability, we attempt smooth control of musical attributes that are novel in the target tasks. To this end, we exploit methods to regularize latent representations in the way they are disentangled by desirable attributes. In particular, we adapt these approaches to a task of generating chord labels from the given melody (Chapter 3) and a task of generating expressive performance parameters from the given musical score (Chapter 4).

The major contributions of this thesis can be summarized as follows:

- **Input encoding framework**: We propose to encode the raw form of input data representations into the embeddings of musically-meaningful units. We use the alignment path to directly map the input data to a sequence of vectors where the information of each vector corresponds to the target unit. This method does not require any complex computation or adding model architecture such as attention modules. Nonetheless, it has been shown that such a simple operation enables the sequential models to capture temporal dependencies in a natural way. It has led to the improved performance of reconstructing structures in both chord and polyphonic piano performance.

- **Generation framework**: We also extend the generation framework in the target CGMC tasks. In the first study, we newly apply the VNMT framework to the melody harmonization task, proposing VTHarm and rVTHarm that use a variational Transformer to discover the connection between two sequences in dynamic length. In the second study, we apply two-step decoders that autoregressively generate the notewise parameters from the chordwise latent representations. The experimental results from the two studies show that these novel approaches can enhance the power of creating musically plausible outputs.

- **Controllable factors**: We have attempted to control novel factors in both CGMC tasks. The controllable factors have been limited to a few attributes such as rhythm or note density. In the first study, we attempt to smoothly control the chord coverage, which has not been deeply investigated in this field, using rVTHarm. In the second study, we propose to control novel attributes of expressive piano performance. Concretely, we extract new inductive biases from the raw data using the domain knowledge in music performance. The extensive evaluation of the proposed model and controlling framework reveal that our systems can generate stable, creative piano performances having musical expression that can be controlled independently of the given musical score.

- **Extensive evaluation with multiple datasets**: We have employed multiple datasets for quantitative or qualitative evaluations of the proposed systems to certify the models' robustness and generality. In the first study, we add a new dataset to the common benchmarking dataset, the HLSD dataset, which the baseline studies have used. This dataset additionally assesses how the models work on the transposed melodies and the data with high musical tension. In the second study, the proposed model is evaluated by various external datasets in addition to the training data from Chopin's pieces. In particular, we evaluate the model with the dataset for various composers and non-Classical music to generalize the model performance. As a result, these various datasets have verified the proposed methods from multiple points of view, proving that they could learn meaningful representations without being overfitted.

5.2 Future Work

Although we have provided some contributions to the CGMC field, our works still have limitations in terms of the following reasons. First, we have not deeply assessed the controllable factors themselves. Despite their novelty and feasibility earned by the literature backgrounds, we need more extensive investigation on correspondence between the factors and the extracted annotations, or the pseudo labels. Second, we need more analysis of the qualitative results. We need to deeply investigate various external variables that can be derived from the collected participants to get clearer insights into the proposed model. Finally, the datasets used in this thesis are still limited to small sizes and small sets of genres or styles. Using shallow datasets may be useful to reduce human labor or reduce some time on preprocessing procedures. However, the generative models can be more improved in their representing power and generality by increasing the scale of the datasets. In this section, we suggest some future directions in detail to mitigate the aforementioned limitations and further develop the CGMC field.

5.2.1 Deeper Investigation of Controllable Factors

We can additionally explore the factors that we already have considered in this thesis or search for more novel factors to expand musical creativity. In the first study, we can arrange more experiments to define the "chord complexity". We have naively defined it as chord coverage which represents the number of unique chords. However, chord complexity is actually entangled with human perception and a number of musical attributes including harmonic rhythm or dissonance [161]. Hence, we may need to further factorize this property into sub-factors and investigate whether these sub-factors can be beneficial for clearer disentanglement of the representations. Moreover, we can also seek other attributes of chord progression such as bass or inversion. In the second study, we can further explore how the "explicit planning" and the remaining "structural attribute" should be clearly represented. We have chosen polynomial regression as a method for extracting the smooth sketch of planning from the performance data. However, this method may have not been helpful for balancing the expressive power of the latent representations for the two factors. We can examine other methods that can more clearly reveal the structural attribute, and discover more abstract attributes of music performance such as a player's style that is independent of the explicit planning.

5.2.2 More Analysis of Qualitative Evaluation Results

It has been evident from this thesis that the musical background can be a significant factor that influences human perception of the generated music. Hence, future studies can arrange additional qualitative tests to discover the correlation of the participant groups with the particular pattern of the results. In the first study, we can investigate how melody awareness is connected to preference and the model type, especially rVTHarm. In the second study, we can deeply explore the reason why the tendency is largely distinctive between musically trained and untrained groups when listening to the generated piano performances. Furthermore, we can also compare the proposed rendering model to the conventional method such as VirtuosoNet by Jeong *et al.* [18]. This may be beneficial to get useful insights into the perceived quality of the generated performances that follow an intention by the composers and the performances that are controlled to violate the intention.

5.2.3 Improving Diversity and Scale of Dataset

Despite the extensive evaluations conducted in this thesis, the genres or styles of the datasets are limited to small ranges. In the first study, we can use Classical or more unique genres other than jazz or pop. In the second study, we can train the model with datasets of various composers that span various eras, and the non-classical datasets for the listening test can be further studied by sub-genres, such as OST, blues, or pop. Furthermore, the size of the datasets can be extended for both tasks. For melody harmonization, we can append polyphonic music datasets where the chord annotations are aligned with single or multiple tracks. Such datasets can be often found in different music generation tasks, such as pop music generation or accompaniment generation, particularly where [39]. For the performance rendering task, we can substantially augment the piano performance dataset by modifying tempo, dynamics, or transpositions [14].

Bibliography

[1] D. Makris, I. Karydis, and S. Sioutas, “Automatic melodic harmonization: An overview, challenges and future directions,” in *Trends in Music Information Seeking, Behavior, and Retrieval for Creativity*, 2016.

[2] A. Bhatara, A. K. Tirovolas, L. M. Duan, B. Levy, and D. J. Levitin, “Perception of emotional expression in musical performance,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 37, no. 3, pp. 921–934, 2011.

[3] G. Widmer and W. Goebl, “Computational models of expressive music performance: The state of the art,” *Journal of New Music Research*, vol. 33, no. 3, pp. 203–216, 2004.

[4] B. L. Jacob, “Algorithmic composition as a model of creativity,” *Organised Sound*, vol. 1, no. 3, pp. 157–165, 1996.

[5] S. Ji, J. Luo, and X. Yang, “A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions,” *arXiv preprint arXiv:2011.06801*, 2020.

[6] J.-P. Briot and F. Pachet, “Deep learning for music generation: Challenges and directions,” *Neural Computing and Applications*, vol. 32, pp. 981–993, 2020.

[7] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, “This time with feeling: Learning expressive musical performance,” *Neural Computing and Applications*, vol. 32, pp. 955–967, 2020.

[8] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.

[9] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, “Deep learning techniques for music generation – A survey,” *arXiv preprint arXiv:1709.01620*, 2017.

[10] E. Waite, “Generating long-term structure in songs and stories,” *Magenta Blog: https://magenta.tensorflow.org/blog/2016/07/15/lookback-rnn-attention-rnn/*, 2016, [Online; accessed 18-July-2022].

[11] Y. Zou, P. Zou, Y. Zhao, K. Zhang, R. Zhang, and X. Wang, “MELONS: Generating melody with long-term structure using Transformers and structure graph,” in *Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing*, Singapore, 2022.

[12] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis, “Combining LSTM and feed forward neural networks for conditional rhythm composition,” in *Proceedings of the 18th International Conference on Engineering Applications of Neural Networks*, Athens, Greece, 2017.

[13] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MIDINet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

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

[15] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, United States, 2018.

[16] C. E. Cancino-Chacón and M. Grachten, "An evaluation of score descriptors combined with non-linear models of expressive dynamics in music," in *Proceedings of the 18th International Conference on Discovery Science*, Alberta, Canada, 2015.

[17] Z. Shi, "Computational analysis and modeling of expressive timing in Chopin Mazurkas," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[18] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "VirtuosoNet: A hierarchical RNN-based system for modeling expressive piano performance," in *Proceedings of the 20th International Society for Music Information Retrieval*, Delft, The Netherlands, 2019.

[19] L. N. Ferreira and J. Whitehead, “Learning to generate music with sentiment,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[20] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, “EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[21] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, “MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018.

[22] H. H. Mao, T. Shin, and G. W. Cottrell, “DeepJ: Style-specific music generation,” in *Proceedings of the 12th International Conference on Semantic Computing*, Laguna Hills, United States, 2018.

[23] J. Ens and P. Pasquier, “MMM: Exploring conditional multi-track music generation with the Transformer,” *arXiv preprint arXiv:2008.06048*, 2020.

[24] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, “Multitrack Music Transformer: Learning long-term dependencies in music with diverse instruments,” *arXiv preprint arXiv:2207.06983*, 2022.

[25] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, “Automatic melody

harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2020.

[26] H. Lim, S. Rhyu, and K. Lee, "Chord generation from symbolic melody using BLSTM networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[27] K.-W. Chen, H.-S. Lee, Y.-H. Chen, and H.-M. Wang, "SurpriseNet: Melody harmonization conditioning on user-controlled surprise contours," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[28] Y. Ren, J. He1, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "PopMAG: Pop music accompaniment generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, Seattle, United States, 2020.

[29] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[30] K. Choi, J. Park, W. Heo, S. Jeon, and J. Park, "Chord conditioned melody generation with Transformer based decoders," *IEEE Access*, vol. 9, pp. 42 071–42 080, 2021, doi: 10.1109/ACCESS.2021.3065831.

[31] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[32] H.-P. Lee, J.-S. Fang, and W.-Y. Ma, “iComposer: An automatic songwriting system for Chinese popular music,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, Minnesota, 2019.

[33] Y. Yu, A. Srivastava, and S. Canales, “Conditional LSTM-GAN for melody generation from lyrics,” *The ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1, pp. 1–20, 2021.

[34] Z. Wang and G. Xia, “MuseBERT: Pre-training of music representation for music understanding and controllable generation,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[35] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, “Encoding musical style with Transformer autoencoders,” in *Proceedings of the 37th International Conference on Machine Learning*, Online, 2020.

[36] I. Simon, D. Morris, and S. Basu, “MySong: Automatic accompaniment generation for vocal melodies,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy, 2008.

[37] H. Zhu, Q. Liu, N. J. Yuan, C. Qin, J. Li, K. Zhang, G. Zhou, F. Wei, Y. Xu, and E. Chen, “Xiaoice Band: A melody and arrangement generation framework for pop music,” in *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, London, United Kingdom, 2018.

[38] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, "Pianotree VAE: Structured representation learning for polyphonic music," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[39] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, Seattle, United States, 2020.

[40] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpainting," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[41] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, no. 25, pp. 1–23, 2018.

[42] C. E. Cancino-Chacón, T. Gadermaier, G. Widmer, and M. Grachten, "An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music," *Machine Learning*, vol. 106, no. 6, pp. 887–909, 2017.

[43] M. Grachten and F. Krebs, "An assessment of learned score features for modeling expressive dynamics in music," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1211–1218, 2014.

[44] S. V. Herwaarden, M. Grachten, and W. B. D. Haas, “Predicting expressive dynamics in piano performances using neural networks,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014.

[45] C. V. Patricio and H. Honing, “Generating expressive timing by combining rhythmic categories and Lindenmayer systems,” in *Proceedings of the 50th Annual Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour*, London, United Kingdom, 2014.

[46] G. Widmer, S. Flossmann, and M. Grachten, “YQX plays Chopin,” *AI Magazine*, vol. 30, no. 3, pp. 35–48, 2009.

[47] D. Jeong, T. Kwon, and J. Nam, “VirtuosoNet: A hierarchical attention RNN for generating expressive piano performance from music score,” in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, Montréal, Canada, 2018.

[48] A. Maezawa, “Deep piano performance rendering with conditional VAE,” in *Late-Breaking Demo, the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018.

[49] J. H. McDermott and A. J. Oxenham, “Music perception, pitch, and the auditory system,” *Current Opinion in Neurobiology*, vol. 18, no. 4, pp. 452–463, 2008.

[50] G. E. Poliner, D. P. W. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, “Melody transcription from music audio: Approaches and evalu-

ation," *The IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.

[51] B. Benward and M. Saker, *Music in theory and practice.* McGraw-Hill Higher Education, 2003, vol. 1.

[52] S. Madjiheurem, L. Qu, and C. Walder, "Chord2Vec: Learning musical chord embeddings," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

[53] T. H. Axel Berndt, "Modelling musical dynamics," in *Proceedings of the 5th Audio Mostly Conference*, New York, United States, 2010.

[54] S. Flossmann, M. Grachten, and G. Widmer, "Expressive performance rendering with probabilistic models," in *Guide to Computing for Expressive Music Performance.* London, United Kingdom: Springer, 2013, pp. 75–98.

[55] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[56] H. Tsushima, E. Nakamura, K. Itoyama, and K. Yoshii, "Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[57] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering music performance with interpretation variations using conditional variational RNN,"

in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[58] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting," in *Proceddings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, 2019.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[60] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modeling," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[61] Y. Li and S. Mandt, "Disentangled sequential autoencoder," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 2018.

[62] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[63] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013.

[64] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, United States, 2019.

[65] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[66] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 2018.

[67] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[68] A. Pati and A. Lerch, "Latent space regularization for explicit control of musical attributes," in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, United States, 2019.

[69] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-song and fine-grained music style transfer with just one Transformer VAE," *arXiv preprint arXiv:2105.04090*, 2021.

[70] J. Zhao and G. Xia, "AccoMontage: Accompaniment arrangement via phrase selection and style transfer," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[71] Y. Zhang, Z. Wang, D. Wang, D. Wang, and G. Xia, "BUTTER: A representation learning framework for bi-directional music-sentence retrieval and generation," in *Proceedings of the 1st Workshop on NLP for Music and Audio*, Montréal, Canada, 2020.

[72] A. Wuerkaixi, C. Benetatos, and C. Z. Zhiyao Duan, "CollageNet: Fusing arbitrary melody and accompaniment into a coherent song," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[73] K. Chen, C.-i. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[74] F. Lerdahl and R. Jackendoff, "An overview of hierarchical structure in music," *Music Perception*, vol. 1, no. 2, pp. 229–252, 1984.

[75] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word Transformer: Learning to compose full-song music over dynamic directed

hypergraphs," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Online, 2021.

[76] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017.

[77] L. Angioloni, T. Borghuis, L. Brusci, and P. Frasconi, "CONLON: A pseudo-song generator based on a new pianoroll, Wasserstein autoencoders, and optimal interpolations," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[78] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning," in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020.

[79] Y. Zhang, X. Gao, S. Lee, C. Brockett, M. Galley, J. Gao, and B. Dolan, "Improving the dialogue generation consistency via self-supervised learning," *arXiv preprint arXiv:1905.12681*, 2019.

[80] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf, "S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, United States, 2020.

[81] A. Pati and A. Lerch, “Attribute-based regularization of latent spaces for variational auto-encoders,” *Neural Computing and Applications*, vol. 33, pp. 4429–4444, 2020.

[82] A. L. Ashis Pati, “Is disentanglement enough? on latent representations for controllable music generation,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[83] M. J. Steedman, “The blues and the abstract truth: Music and mental models,” *Mental Models in Cognitive Science*, pp. 305–318, 1996.

[84] J.-F. Paiement, D. Eck, and S. Bengio, “Probabilistic melodic harmonization,” in *Proceedings of the 19th Conference of the Canadian Society for Computational Studies of Intelligence*, Québec City, Canada, 2006.

[85] P. R. Illescas, D. Rizo, and J. M. Iñesta, “Harmonic, melodic, and functional automatic analysis,” in *Proceedings of the 33rd International Computer Music Conference*, Copenhagen, Denmark, 2007.

[86] H. V. Koops, J. P. Magalhães, and W. B. de Haas, “A functional approach to automatic melody harmonisation,” in *Proceedings of the 1st ACM SIGPLAN Workshop on Functional Art, Music, Modelling and Design*, New York, United States, 2013.

[87] M. Tokumaru, K. Yamashita, N. Muranaka, and S. Imanishi, “Membership functions in automatic harmonization system,” in *Proceedings of the 28th IEEE International Symposium on Multiple-Valued Logic*, Fukuoka, Japan, 1998.

[88] A. R. R. Freitas and F. G. Guimarães, "Melody harmonization in evolutionary music using multiobjective genetic algorithms," in *Proceedings of the 8th Sound and Music Computing Conference*, Padova, Italy, 2011.

[89] M. Kaliakatsos–Papakostas and E. Cambouropoulos, "Probabilistic harmonization with fixed intermediate chord constraints," in *Proceedings of the 40th International Computer Music Conference*, Athens, Greece, 2014.

[90] H. Tsushima, E. Nakamura, and K. Yoshii, "Bayesian melody harmonization based on a tree-structured generative model of chord sequences and melodies," *IEEE/ACM Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 1644–1655, 2020.

[91] C.-E. Sun, Y.-W. Chen, H.-S. Lee, Y.-H. Chen, and H.-M. Wang, "Melody harmonization using orderless NADE, chord balancing, and blocked Gibbs sampling," in *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, 2021.

[92] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the KTH rule system for musical performance," *Advances in Cognitive Psychology*, vol. 2, no. 2-3, pp. 145–161, 2006.

[93] A. Friberg and E. E. Bisesi, "Using computational models of music performance to model stylistic variations," *Expressiveness in music performance: Empirical approaches across styles and cultures*, pp. 240–259, 2014.

[94] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," *Artificial Intelligence*, vol. 146, no. 2, pp. 129–148, 2003.

[95] K. Teramura, H. Okuma, Y. Taniguchi, S. Makimoto, and S. ichi Maeda, "Gaussian process regression for rendering music performance," in *Proceedings of the 10th International Conference on Music Perception and Cognition*, Sapporo, Japan, 2008.

[96] Y. Gu and C. Raphael, "Modeling piano interpretation using switching kalman filter," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.

[97] T. H. Kim, S. Fukayama, T. Nishimoto, and S. Sagayama, "Statistical approach to automatic expressive rendition of polyphonic piano music," in *Guide to Computing for Expressive Music Performance.* London, United Kingdom: Springer, 2013, pp. 145–179.

[98] S. Tanaka, M. Hashida, and H. Katayose, "Shunji: A case-based performance rendering system attached importance to phrase expression," in *Proceedings of the 8th Sound and Music Computing Conference*, Padova, Italy, 2011.

[99] S. Lauly, "Modélisation de l'interprétation des pianistes and applications d'auto-encodeurs sur des modèles temporels," Master's thesis, University of Montréal, 2010.

[100] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proceedings*

of the 36th International Conference on Machine Learning, Long Beach, United States, 2019.

[101] J. Wu, C. Hu, Y. Wang, X. Hu, and J. Zhu, “A hierarchical recurrent neural network for symbolic melody generation,” *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2749–2757, 2019.

[102] I. Simon, A. Roberts, C. Raffel, J. Engel, C. Hawthorne, and D. Eck, “Learning a latent space of multitrack measures,” in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, Montréal, Canada, 2018.

[103] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, “LakhNES: Improving multi-instrumental music generation with cross-domain pre-training,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[104] S.-L. Wu and Y.-H. Yang, “The Jazz Transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[105] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, “Hierarchical generative modeling for controllable speech synthesis,” in *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, United States, 2019.

[106] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong, "Disentangled and controllable face image generation via 3D imitative-contrastive learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, United States, 2020.

[107] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[108] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in VAEs," in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, Montréal, Canada, 2018.

[109] H. Kim and A. Mnih, "Disentangling by factorising," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, 2018.

[110] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

[111] E. Battenberg, S. Mariooryad, D. Stanton, R. Skerry-Ryan, M. Shannon, D. Kao, and T. Bagby, "Effective use of variational embedding capacity in expressive end-to-end speech synthesis," *arXiv preprint arXiv:1906.03402*, 2019.

[112] S. Tulyakov, M. Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, United States, 2018.

[113] S. Reed, K. Sohn, Y. Zhang, and H. Lee, "Learning to disentangle factors of variation with manifold interaction," in *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014.

[114] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: A deep model for learning face identity and view representations," in *Proceedings of the 28th Conference on Neural Information Processing Systems*, Montréal, Canada, 2014.

[115] S. Narayanaswamy, T. B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[116] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, Barcelona, Spain, 2016.

[117] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network. in advances in neural information

processing systems," in *Proceedings of the 29th Conference on Neural Information Processing Systems*, Montréal, Canada, 2015.

[118] J. Chen and K. Batmanghelich, "Weakly supervised disentanglement by pairwise similarities," in *Proceedings of The 34th AAAI Conference on Artificial Intelligence*, New York, United States, 2020.

[119] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017.

[120] R. Habib, S. Mariooryad, M. Shannon, E. Battenberg, R. Skerry-Ryan, D. Stanton, D. Kao, and T. Bagby, "Semi-supervised generative modeling for controllable speech synthesis," in *Proceedings of the 8th International Conference on Learning Representations*, Online, 2020.

[121] G. Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Principles for learning controllable TTS from annotated and latent variation," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, 2017.

[122] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, United States, 2017.

[123] Y. Lee, R. Azam, and S.-Y. Lee, "Emotional end-to-end neural speech synthesizer," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[124] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, 2019.

[125] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understandings," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, United States, 2017.

[126] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proceedings of the 14th European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016.

[127] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Santiago, Chile, 2015.

[128] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[129] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, United States, 2019.

[130] O. Wiles, A. S. Koepke, and A. Zisserman, “X2Face: A network for controlling face generation by using images, audio, and pose codes,” in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 2018.

[131] A. Nagrani, J. S. Chung, S. Albanie, and A. Zisserman, “Disentangled speech embeddings using cross-modal self-supervision,” in *Proceedings of the 45th IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020.

[132] A. Rouditchenko, H. Zhao, H. Gan, J. McDermott, and A. Torralba, “Self-supervised audio-visual co-segmentation,” in *Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, United Kingdom, 2019.

[133] K. Okumura, S. Sako, and T. Kitamura, “Stochastic modeling of a musical performance with expressive representations from the musical score,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, United States, 2011.

[134] ——, “Laminae: A stochastic modeling-based autonomous performance rendering system that elucidates performer characteristics,” in *Proceedings of the 40th International Computer Music Conference*, Athens, Greece, 2014.

[135] E. Stamatatos and G. Widmer, “Automatic identification of music performers with learning ensembles,” *Artificial Intelligence*, vol. 165, 2005.

[136] B. Gingras, P.-Y. Asselin, and S. McAdams, “Individuality in harpsichord performance: Disentangling performer- and piece-specific influences on interpretive choices,” *Frontiers in Psychology*, vol. 4, 2013.

[137] H. H. Tan, Y.-J. Luo, and D. Herremans, “Generative modeling for controllable audio synthesis of expressive piano performance,” in *Proceedings of the 37th International Conference on Machine Learning*, Online, 2020.

[138] S. A. Raczyński, S. Fukayama, and E. Vincent, “Melody harmonization with interpolated probabilistic models,” *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[139] C. Anderson, D. Carlton, R. Miyakawa, and D. Schwachhofer. (2021) Hooktheory. Accessed: 2021-09-05. [Online]. Available: https://www.hooktheory.com

[140] S. Hiehn. (2019) Chord Melody Dataset. Accessed: 2021-09-05. [Online]. Available: https://github.com/shiehn/chord-melody-dataset

[141] H. C. Longuet-Higgins and M. J. Steedman, “On interpreting Bach,” *Machine Intelligence*, vol. 6, pp. 221–241, 1971.

[142] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[143] Z. Lin, G. I. Winata, P. Xu, Z. Liu, and P. Fung, “Variational transformers for diverse response generation,” *arXiv preprint arXiv:2003.12738*, 2020.

[144] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang, “Variational neural machine translation,” in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, United States, 2016.

[145] X. Sheng, L. Xu, J. Guo, J. Liu, R. Zhao, and Y. Xu, "IntroVNMT: An introspective model for variational neural machine translation," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, United States, 2020.

[146] X. Liu, J. Zhao, S. Sun, H. Liu, and H. Yang, "Variational multimodal machine translation with underlying semantic alignment," *Information Fusion*, vol. 69, pp. 73–80, 2021.

[147] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural speech synthesis with Transformer network," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, United States, 2019.

[148] C. Raphael and J. Stoddard, "Functional harmonic analysis using probabilistic models," *Computer Music Journal*, vol. 28, no. 3, pp. 45–52, 2004.

[149] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proceedings of the 29th Conference on Neural Information Processing Systems*, Montréal, Canada, 2015.

[150] A. Roberts and C. F. Hawthorne. (2021) Magenta musicXML parser. Accessed: 2021-09-05. [Online]. Available: https://github.com/magenta/note-seq/blob/main/note_seq/musicxml_parser.py

[151] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[152] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in *Pro-*

ceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, United States, 2018.

[153] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, United States, 2019.

[154] T. Fujishima, "Realtime chord recognition of musical sound: A system using Common Lisp Music," in *Proceedings of the 25th International Computer Music Conference*, Beijing, China, 1999.

[155] C. A. Harte, M. B. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of the 1st ACM International Conference on Multimedia*, Santa Barbara, United States, 2006.

[156] J.-F. Paiement, D. Eck, and S. Bengio, "A probabilistic model for chord progressions," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, London, United Kingdom, 2005.

[157] W. B. de Haas, F. Wiering, and R. C. Veltkamp, "A geometrical distance measure for determining the similarity of musical harmony," *International Journal of Multimedia Information Retrieval*, vol. 2, pp. 189–202, 2013.

[158] F. Lerdahl, "Tonal pitch space," *Music Perception*, vol. 5, no. 3, pp. 315–349, 1988.

[159] E. Cambouropoulos, “A directional interval class representation of chord transitions,” in *Proceedings of the 12th International Conference on Music Perception and Cognition*, Thessaloniki, Greece, 2012.

[160] A. Marsden, “Schenkerian analysis by computer: A proof of concept,” *Journal of New Music Research*, vol. 39, no. 3, pp. 269–289, 2010.

[161] B. D. Giorgi, S. Dixon, M. Zanoni, and A. Sarti, “A data-driven model of tonal chord sequence complexity,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2237–2250, 2017.

[162] R. Bresin and A. Friberg, “Emotional coloring of computer-controlled music performances,” *Computer Music Journal*, vol. 24, 2000.

[163] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson, “Changing musical emotion: A computational rule system for modifying score and performance,” *Computer Music Journal*, vol. 34, 2010.

[164] M. Bernays and C. Traube, “Investigating pianists’ individuality in the performance of five timbral nuances through patterns of articulation, touch, dynamics, and pedaling,” *Frontiers in Psychology*, vol. 5, no. 157, pp. 1–19, 2014.

[165] R. H. Woody, “The relationship between explicit planning and expressive performance of dynamic variations in an aural modeling task,” *Journal of Research in Music Education*, vol. 47, no. 4, pp. 331–342, 1999.

[166] A. Lerch, C. Arthur, A. Pati, and S. Gururani, “Music performance analysis: A survey,” in *Proceedings of the 20st International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[167] B. H. Repp, "A microcosm of musical expression: II. Quantitative analysis of pianists' dynamics in the initial measures of Chopin's Etude in E major," *The Journal of the Acoustical Society of America*, vol. 105, pp. 1972–1988, 1999.

[168] H. Honing, "From time to time: The representation of timing and tempo," *Computer Music Journal*, vol. 25, 2001.

[169] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Score and performance features for rendering expressive music performances," in *Proceedings of the Music Encoding Conference*, Vienna, Austria, 2019.

[170] A. Roberts, J. Engel, and D. Eck, "Hierarchical variational autoencoders for music," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, United States, 2017.

[171] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proceedings of the 29th Conference on Neural Information Processing Systems*, Montréal, Canada, 2015.

[172] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, 1989.

[173] W. Goebl, "Melody lead in piano performance: Expressive device or artifact?" *The Journal of the Acoustical Society of America*, vol. 110, 2001.

[174] M. Grachten and G. Widmer, "Linear basis models for prediction and analysis of musical expression," *Journal of New Music Research*, vol. 41, no. 4, pp. 311–322, 2012.

[175] F. Foscarin, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: A dataset of aligned scores and performances for piano transcription,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[176] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, United States, 2019.

[177] M. B. Holbrook and P. Anand, “Effects of tempo and situational arousal on the listener’s perceptual and affective responses to music,” *Psychology of Music*, vol. 18, pp. 150–162, 1990.

[178] S. B. Kamenetsky, D. S. Hill, and S. E. Trehub, “Effect of tempo and dynamics on the perception of emotion in music,” *Psychology of Music*, vol. 25, pp. 149–160, 1997.

초 록

음악적 요소를 조건부 생성하는 분야인 CGMC는 멜로디나 화음과 같은 음악의 일부분을 기반으로 나머지 부분을 생성하는 것을 목표로 한다. 이 분야는 음악적 요소 간 복잡한 관계를 탐구하는 데 용이하고, 음악을 만드는 데 어려움을 겪는 비전문가들을 도울 수 있다. 최근 연구들은 딥러닝 기술을 활용하여 CGMC 시스템의 성능을 높여왔다. 하지만, 이러한 연구들에는 아직 생성 품질과 제어가능성 측면에서 두 가지의 한계점이 있다. 먼저, 생성된 음악의 음악적 구조가 명확하지 않다. 또한, 아직 좁은 범위의 음악적 요소 및 테스크만이 유연한 제어의 대상으로서 탐구되었다. 이에 본 학위논문에서는 CGMC의 개선을 위해 위 두 가지의 한계점을 해결하고자 한다. 첫 번째로, 음악 구조를 이루는 음악적 위계를 직관적으로 모델링하는 데 집중하고자 한다. 본래 데이터와 음, 화음과 같은 음악적 단위 간 정렬 경로를 사용하여 모델이 음악적으로 의미있는 종속성을 명확하게 배울 수 있도록 한다. 두 번째로, 잠재 표상을 활용하여 새로운 음악적 요소들을 유연하게 제어하고자 한다. 특히 잠재 표상이 의도된 요소에 대해 풀리도록 훈련하기 위해서 비지도 혹은 자가지도 학습 프레임워크을 사용하여 잠재 표상을 제한하도록 한다. 본 학위논문에서는 CGMC 분야의 대표적인 두 테스크인 멜로디 하모나이제이션 및 표현적 연주 렌더링 테스크에 대해 위의 두 가지 방법론을 검증한다. 다양한 실험적 결과들을 통해 제안한 방법론이 CGMC 시스템의 음악적 창의성을 안정적인 생성 품질로 확장할 수 있다는 가능성을 시사한다.

주요어: 음악 생성, 화음 생성, 연주 생성, 음악 구조, 잠재 표현 학습

학 번: 2015-31346