# From Lyric to Melody: Cross Modal Generation with Two-Stage Modeling

RGDL2
September 2022

A Dissertation submitted in part fulfilment of the
Degree of Master of Science in Machine Learning

# Abstract

Lyric-based melody generation is always an interesting and attractive challenge in the area of automatic songwriting. However, this cross modal generation task can be intractable due to the limitation of lyric-melody paired data and the huge different in modalities between lyric, which is in text modality, and melody, which is in music modality. In this dissertation, we propose a two-stage modeling approach, that utilize pitch as a bridge to link the modality difference between lyric and melody, to simplify the complexity of this task by solving its sub-tasks which can be considered as under single modality respectively. Meanwhile, this thesis contributes to the research and exploration of four hypotheses in the task of lyric-to-melody generation:

1. The feasibility of using phonemes instead of words in the task of lyric-to-melody generation.
2. The effectiveness of utilizing pitches as the bridge to connect lyrics and melodies for the two-stage modeling approach.
3. The availability of our two-stage modeling method in the low resource situation of limited lyric-melody paired data.
4. The validity of integrating domain knowledge in music to our modeling approach that used for melody generation.

Both of the subjective results and objective results from the experiment indicate our method is able to be competent in generating harmonious and natural melodies from given lyrics.

# Acknowledgement

Many thanks to my supervisors, Dr. Tommaso Pasini from Huawei Technologies; Dr. Jun Wang, who is also my personal tutor at University College London, and Dr. Yali Du from Kings College London. Additionally, I want to thank Dr. Jinhua Du who offer me the opportunity to research this both fascinating and challenging area with Huawei. Thanks also to Levan Bui, one of the most important girls during my life, I love every moment spent with you. Last but not least, thank you for your unconditional love and support, mother and father.

# Declaration of Authenticity

This report is submitted as part requirement for the MSc Machine Learning at University College London. It is substantially the result of my own work except where explicitly indicated in the text.

The report will be distributed to the internal and external examiners, but thereafter may not be copied or distributed except with permission from the author.

# Contents

# List of Figures

# List of Tables

# Chapter I

# Introduction

## 1 Lyric-based Songwriting

From the rule-based vowel-to-pitch algorithm devised by Guido D'Arezzo in the 11th century [1] and 18th century Mozart's attempts of Musikalisches Wurfelspiel to compose musical scores through the randomness of dice rolling [2], the research community has never stopped exploring algorithms for automatic music generation. In the field of automatic music generation, there is no doubt that lyric-based songwriting is one of the most interesting and challenging directions: by simply providing the lyrics, the algorithm can generate the melodies associated with it.

The history of songwriting based on lyrics goes back a long way, even 3,000 years. As early as the 11th to 7th centuries B.C., during the period of China's Zhou dynasty, people wrote down the stories of their lives in the form of poems, added melodies and sang them to others in the form of songs [3]. These poems are rich in content, reflecting the Zhou dynasty society's labor and love, war and servitude, oppression and rebellion, customs and banquets, and even astronomical phenomena, geography, animals and plants in that age [4]. Subsequent scholars collected and edited these poems into a general collection, which is one of the cultural treasures of China, the *Classic of Poetry (Shijing)* [5].

In the Middle Ages, minstrels often wandered from tavern to tavern and from the banquets of the nobility to express their own ideas about the world or their feelings about a certain person or event, using stories that actually happened or were created by themselves [6]. In their writing, knights in the face of tens of times the enemy still fought bravely and won the battle, the princess hanging tears to mourn her handsome lover on the terrace under the moon, young men and women hiding among the flowers and dating secretly. All of these were conveyed in a more relax and more accessible way, singing: minstrels made up melodies for the poems and stories and sang the song during the performance [7].

Today, at any Premier League match, the field is filled with fans chanting about lyrics written based on their team and their opponents [8]. For examples, *"Stand up for Arsenal"* at Emirates Stadium, *"We are live in Red and White Kop"* at Anfield, also *"Flying high, up in the sky"* at Stamford Bridge, and of course, *"Harry Maguire, your defence is terrifying"* at Old Trafford [9].

Although it is not hard for anyone to sing a melody by changing the intonation and duration of each word from a given text, but generally, creating melodies that doesn't make audience feel boring for written lyrics manually requires a great deal of knowledge about music theory and training to develop musicality, which can take a long time to learn and understand systematically [10][11]. The advent of songwriting techniques based on lyrics has greatly lowered the threshold for songwriting, allowing anyone without a musical background to write lyrics and create songs using their own ideas, enabling them to enter the wide world of music composition.

# 2 Music Representations

In music, **note** is the most basic unit of composition. The term note is typically used in quite flexible approach and can refer to both a musical notation and a pitched sound. Every note contains certain properties which define the relative **pitch** and **duration** of a sound in a segment of music. Take guitar as an instance, the pitch of a note informs a performer which string to pluck on the acoustic bridge, and the note's duration indicates how long this string is to be kept down [12].

Music is regular and varied repetition, and these various repetitions are usually considered as temporal units which are composed by various of notes, called **beats**.

Among beats, the **bar** is a single unit of time determined through a specific number of it. Separated music into bars not merely reflect the natural rhythms of it, but also give a consistent reference for its composition. The **time signature** indicates the key of the piece of bar, expressed the temporal structure in the music composition. Time signature refers to the pattern of beats, in general speaking, it indicates how many beats there are in each measure. Take $\frac{4}{4}$, which is one of the commonest beat in pop music, as an example, it refers to taking a quarter note as a beat with four beats each bar. Also as another element of temporal structure, **tempo** indicates the speed of those bars.

Music is able to be represented in variety of approaches. In general, there are three main categories are frequently used for representing musical information: **Sheet Music Representation**, **Audio Representation**, and **Symbolic Representation**, while Figure I.1 briefly depicting these three different type of format [13]:

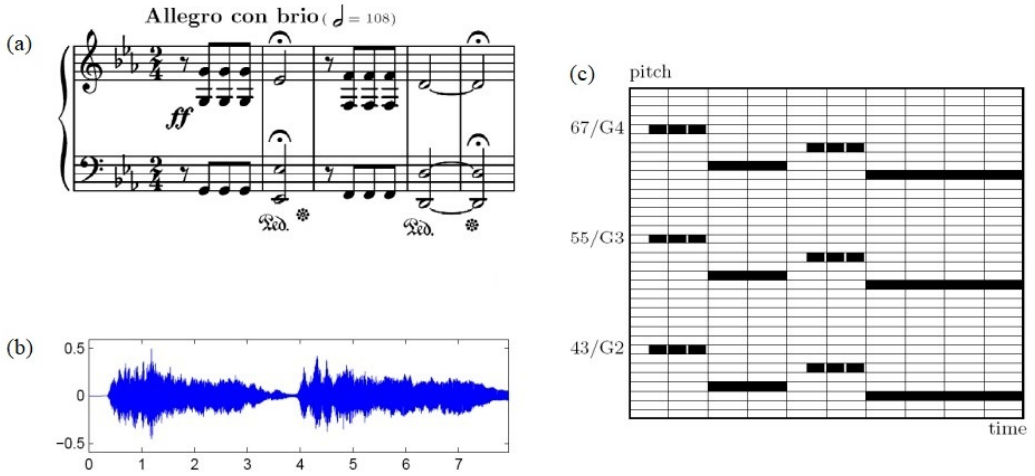

**Figure I.1** ***Music Representations:*** (a) shows the format for sheet music, (b) gives an example of piece of sound with audio representation, and (c) prevents a kind of symbolic representation

**Sheet Music Representation** provides the information of musical score in form of visual description. Normally, it is presented as an image format by either printed or scanned. Sheet music is normally not played robotically. By adjusting the tempo, dynamics, and articulation, performers can influence the flow of the melodies, resulting in a subjective interpretation of the given sheet music. Rather than providing a strict standard, sheet music is able to be viewed as a reference for playing a snatch of music, giving musicians freedom for their personal performance.

**Audio Representation**, on the other hand, presents musical information with a more physical method: through encoding the sound of wave, it can reproduce the acoustic implementation of that piece of music. However, the parameters of musical note are generally implicitly in the audio: neither pitches nor durations are directly indicted in this form of representation. And this makes the task become very tough for music analysis and comparison, especially for music which is superimposed by various type of instruments and sounds like symphony or remix [14].

**Symbolic Representation** is computer parseable that represents as digital format, it describes information in music based on means of entities with explicit musical meaning. Musical Instrument Digital Interface (MIDI) is the most representative and important form of symbolic representation [15]. One of the advantage characters of MIDI format is that it can represent both the physical time of the music like audio representation and the duration of notes. Also, similar to the form of sheet music, MIDI can indicate temporal information using music entities rather than absolute unites of time such as milliseconds. These properties have made MIDI widely used in digital music processing over the past three decades [16].

# 3 Motivations

The lyric-based automatic songwriting as writing melodies from given lyrics belong to multimodal task, as cross modal generation. Although the format of lyric data and melody data are sharing some similarity as both of them can be expressed as sequential discrete token representation, however, their pattern and distribution are different, as in the content lyric data is in form of text while melody data presents music information [17], which causes the lyric and melody are weakly correlated in cross modal generation task.

On the other hand, some of the characteristics in the process of lyrics to melodies songwriting are also distinctive: different with other cross modal generation problems like image to text, every syllable or word in lyrics must has at least one unique strict melody aligned, while other tasks do not have such strictly requirements. Besides, not only pitch and duration, structural information such as beat is also important in a song, since it can directly affect the rhythm in the music [18], and these cause the task becomes more challenging.

Different from other cross modal generation tasks such as image-to-text and text-to-image which have dozens of GB-level modality-paired dataset, for instance the pre-trained multimodal model M6 [19], developed by Alibaba Group and Tsinghua University, using more than size of 1.9TB open source images and 292GB text from internet for their pre-training process, due to the reason of copyright it is intractable to collect million but even 100 thousands of lyric-melody paired song data [20]. Rather than using existing well paired song dataset, there are another idea that tried to extract lyrics and their paired melodies from songs which they crawled from websites such as Facebook and YouTube [21]. However, including recognize and separate human voices from a multi track music, also pair each pronounce of voice to their according lyrics, such data process can also cause the requirement for massive of the computational resources [22].

Hence, exploring a data efficient method for cross modal generation is meaningful, especially at the situation that can exploit current pre-trained models.

# 4 Objectives

The main objective of this thesis is to explore and implement a state-of-the-art method for the generation task of lyrics in Chinese to melodies automatic songwriting under low resources. More specifically, we would like to address the following four main hypothesis in this dissertation:

1. There are more than 90000 characters in Chinese, but their pronounces can be replaced with combinations of 58 phonemes. Using phonemes instead of characters in the lyrics, lots of deep learning based model for singing voice synthesis (SVS) task in Chinese have already achieved impressive results, with audio as their target modality [23][24][25]. Can we also have an

outperformance by utilizing phonemes to substitute for words in Chinese lyric to generate melodies in format of symbolic representation?

2. The direct correlation between lyrics and melodies are relatively low since the huge difference between text and music modality, while the relationship between pitches and phonemes or melodies is more intuitive. If it is possible for us to design a two-stage model with pitch as the bridge that can be more effectively for the lyric-to-melody cross modal generation task?

3. For our designed two-stage generative model, can we exploit the existing pre-training model for each module, through the power of transfer learning, to have an advanced generation performance under the situation of low-resource in limited lyric-melody paired data?

4. Under the end-to-end deep learning based framework, if it is possible for us to integrate principles from domain of music theory to make the neural model demonstrate more superiority in the task of lyric-conditioned melody generation?

## 5 Structure of the Thesis

This dissertation is organized in the following parts:

- **Chapter 2** presents a background overview related to the problem of lyric-to-melody cross-model generation. We firstly describe a literature review of the task. Then, principles of generative models and relevant work in automatic songwriting are presented;

- **Chapter 3** provides our novel approaches for the lyric-to-melody generation task. At the beginning we propose our idea and method about the 2-stage modeling for multi-modal generation, and following that we further describe the technical details about models we used for the stage one and stage two of our generation system;

- **Chapter 4** covers the procedure of the experiment. The dataset we processed for experiment is introduced at first. We next discuss the training details for both the modules in our lyric-to-melody system. Also, configurations for our baseline models are proposed. After that, we further describe our evaluation metrics about the quantitative measuring of generated melodies. And finally, experiment results are provided, also the further analysis and finding among results are presented;

- **Chapter 5** summaries the thoughts related to this project, and discussing future possible directions for the further research.

# Chapter II

# Background

## 1 Multimodal Machine Learning

Modality is an abstract concept that refer to the way of expression or perception of things. Take an example, image is belong to the modality of visual. While similarly, video with sound can be treated as multimodality of visual and audio: qualities, structures and representations are all heterogeneous for the expressions of information in these different modalities, but they also have interconnections within each other. The multimodal problems are widely exist in real-word applications, including but not limited to drug development [26], clinical medicine [27], autonomous driving [28], recommender systems [29], and search engines [30]. In such a challenging and meaningful area, multimodal machine learning is a machine learning paradigm in artificial intelligence that specifically involves using computer algorithms to interact and model multiple sources of heterogeneous and interrelated by computer algorithms through learned experience from data which is in form of multiple modalities.

According to the widely accepted definition [31][32], there are 6 core challenges in the field of multimodal machine learning, while figure II.1 [32] describes the relationship among these tasks:

- **Representation**: Learning representations from multiple different modalities to express the cross-modal interactions among each separate modalities. The sub-challenges of Representation including **Fusion**, which is to learn a computational efficient joint representation across each separate elements of various modalities that reflect their cross-modal interactions; **Coordination** as learning the representations from multiple different modalities contextually which are regulated by their cross-modal interactions so that improving the capability of multimodal contextualization; and **Fission** for learning a new disjoint representation collections, for example clustering or data factorization, which are able to present knowledge of multimodal internal structure

- **Alignment**: Using the data structure and cross-modal interactions to identify and model the connection between all elements among multiple different modalities The sub-challenges in Alignment are categorized into **Connections**, for recognizing the connection between each part of the component among different modalities; **Aligned Representations**, which means to learn a better representations through modeling the cross-modal connections and interactions of every elements; and **Segmentation** which is to deal with the exist of element's granularity and ambiguity in segmentation in the process of alignment
- **Reasoning**: Combining knowledge of evidences in form of multiple modalities to deal with some specific task by utilizing multimodal alignment and problem structure through one or multiple steps of inference. Reasoning contains 4 sub-challenges. **Structure Modeling** is to define or learn the relationships over which reasoning occurs; **Intermediate Concepts** characterizes the parameterization during the reasoning process for individual multimodal concepts; **Inference Paradigm** is about the exploration in infer more abstract concepts from a single evidence in form of multimodal; and **External Knowledge** as studying multimodal structure, concepts, and inference with the leverage of external knowledge
- **Generation**: learning to generate the content which is able to reflect interactions, structure, and coherence in cross-modal in the form of raw modalities The main 3 sub-challenges in Generation are **Translation**, it's about from one modality transfer to another, and keeping the consistent with cross-modal interaction under the condition of maintaining content of

information; **Summarization** as summarizing multimodal data by hightling the most significant part and omit the unimportant content so that streamline the input information; **Creation** for generating multiple modalities at the same time in order to increase the content of the information, and meanwhile maintaining the consistency of internal-modal and cross-modal

- **Transference**: Transfer knowledge from modality to modality as well as their representations, using the modality with sufficient resources to help the primary modality which with limited resources or might be noisy. Sub-challenges among Transference contains **Transfer** for transferring knowledge from large-scale pretrained models to specific downstream tasks involving the primary modality; **Representation enrichment** is transferring knowledge from secondary to primary modality by a joint model sharing representation spaces among these modalities; and similarly, **Model Induction** is transferring knowledge from model in secondary modality to primary modal model while staying their own modality separately;

- **Quantification**: Empirical and theoretical study to better understand the performance of multimodal machine learning. Quantification focuses in **Heterogeneity**, which measuring the dimensions of heterogeneity for multimodal data and investigate how they affect modeling and learning in multimodal machine learning, and these dimensions including but not limited to Structure, Relevance, Precision, Representation space, Information and Noise; **Cross-modal Interactions** is quantifying the form and paradigm for the connections of cross-modal and interactions in data and models of multimodal; and **Multimodal Learning Process** represents to handle problems exist in learning and optimization during the process of learning from heterogeneous data.

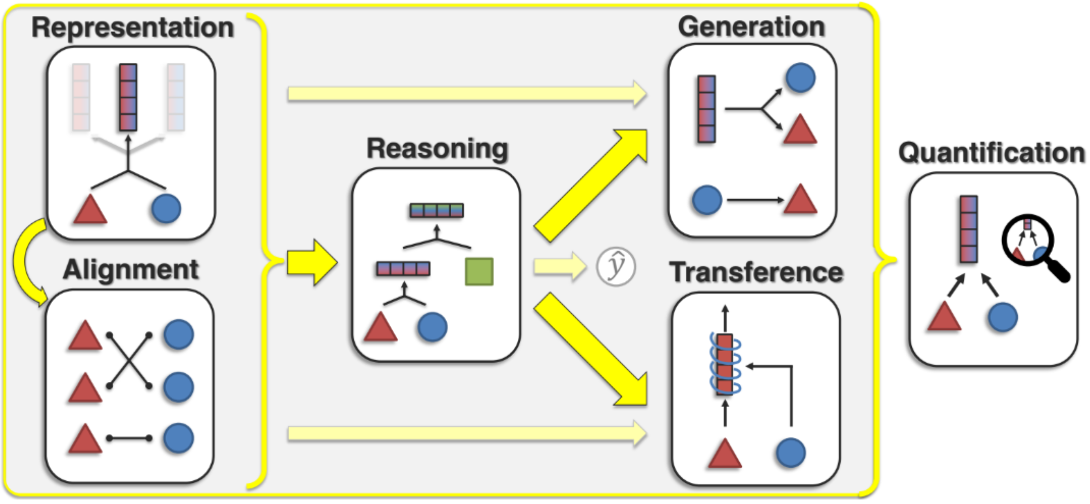

**Figure II.1** *The relationship between each challenge in multimodal machine learning*

## 2 Multimodal Translation

Lyric to melody Cross-modal generation belongs to the generation task of the 6 core challenges of multimodal machine learning. And more specifically, the multimodal translation task of the multimodal generation task.

The purpose of multimodal translation is to learn an approach so that able to map the source modality to the target modality. For example, generating a relevant image from the description of an input sentence [33], or summarizing an input video with a paragraph [34]. Although the methods for multimodal translation are extensive and are usually modality specific, based on similar characteristics these can be divided as two main categories of the paradigm for multimodal translation: *example-based*, and *generative* [31].

Example-based models transfer one modality to another by a constructed dictionary. In the past, the generative model is challenging to construct because it needs to generate a structurally consistent signals or sequential symbols. And therefore, most of the multimodal translation systems in the early stage depends on example-based translation [35]. Example-based algorithms directly build the dictionary from the training dataset [36], while generally they make a translation between modalities through either doing a retrieval within whole dictionary without modifying it [37][38] or creating a set of complex rules to combine several retrieved instances together [39][40].

Generative approaches, with a different thought, develop a model which can generate translated result in the form of target modality rather than retrieving from data. Besides the generative models used for multimodal translation need to understand the representations for both of the source and the target modality, they also requires the capability to model the distribution of the data. A generative model is to model a conditional probability for observed dataset $\mathbf{x} \in \mathcal{X}$ from given target dataset $\mathbf{y} \in \mathcal{Y}$ [41]. Through modeling the joint probability for observable source data $\mathbf{x}$ and target data $\mathbf{y}$, then the generative model can utilize conditional probability function to compute $p(\mathbf{y}|\mathbf{x})$:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \end{aligned} \tag{II.1}$$

Among the equation II.1, $p(\mathbf{y})$ is the prior probability of target dataset $\mathbf{y}$, it describes the probability distribution of $\mathbf{y}$ without knowing the feature $\mathbf{x}$. From training dataset the model is aiming to learn the prior $\mathbf{y}$ and the conditional distribution $p(\mathbf{x}|\mathbf{y})$, then the joint probability distribution $p(\mathbf{x},\mathbf{y})$ can be calculated through measuring the posterior distribution $p(\mathbf{y}|\mathbf{x})$ [42]. For the purpose of generation, the generative model is designed to learn the probability distribution $p(\mathbf{x})$ over $\mathbf{x}$ by density estimation to $\mathbf{x}$, so that it can generate new data $\mathbf{x}_{new}$ through sampling from $p(\mathbf{x})$ [43].

However, example-based multimodal translation methods having some serious issues. One is that the inference speed is relative slow due to the large size of the multimodalities-paired dictionary [44], while another problem is that once the task is not simple enough, then it's not practical to achieve an accurate and contextual translation which is related to source instance based on the trained dictionary [45]. At the advent of the era of deep learning, these methods have largely been replaced by deep generative models, which are generative models with deep neural networks, no matter for multimodal translation tasks like Text-Image [46], Video-Audio [47], Video-Text [48] and broadly other areas including songwriting.

# 3 Deep Generative Modeling

This section involves basic background of deep generative models. It is not designed to be a comprehensive overview of deep generative models, but rather a brief introduction to the principles of the widely used deep generative modelling techniques that can be applied in the area of automatic songwriting.

The objective of deep generative models is about modeling a distribution as close as possible to the true data distribution $p(\mathbf{x})$, which can be solved by maximizing maximum likelihood [49]. This is extremely difficult especially when the real data distribution is high dimensional, such as the form like text, image, and audio [50]. From the point of view of dealing with the maximum likelihood estimation to $p(\mathbf{x})$, the deep generative models can be divided into the following types:

- **Autoregressive models** in 3.1 decompose the objective function into the form of product of conditional probability;
- **Flow models** in 3.2 calculate likelihood function directly by invertible nonlinear transformation techniques;
- **Latent Variable Models** such as Variational Autoencoders (VAEs) in 3.3 are designed for avoiding the probability calculation through maximizing the lower bound of data's log-likelihood;
- **Implicit Models** like Generative Adversarial Networks (GANs) in 3.4 aims to accomplish an equilibrium between generator and discriminator by adversarial training in order to avoid to learn the exact data distribution;
- **Energy-based Models** in 3.5 are inspired from concepts in world of physics, using an alternative function instead of maximizing likelihood directly.

## 3.1 Autoregressive Models

Autoregressive is derived from statistical methods dealing with time series, it predicts the observation value of the current moment with the observation value of each moment before the same variable [51]. Any model that expresses the relationship between adjacent elements of visible data in terms of conditional probabilities and expresses the joint probability distribution as a product of conditional probabilities can be called an autoregressive model [52].

The idea of autoregressive models is to represent the distribution over $p(\mathbf{x})$ sequentially with an autoregressive way: by modeling the distribution $p(\mathbf{x}_i|\mathbf{x}_{<i})$, while **i** is the index in form of discrete in the range of 0 to the number of samples $d$, then the joint probability over $\mathbf{x}$ can be factored in to the form of conditional probability through the chain rule for probability, as equation II.2:

$$p(\mathbf{x}) = \prod_{i=1}^{d} p(\mathbf{x}_i|\mathbf{x}_{<i}) \tag{II.2}$$

Through iteratively sampling $\mathbf{x}_i$ from $p(\cdot|\mathbf{x}_{<i})$, the autoregressive model can hence learn the probability distribution of $\mathbf{x}$ under the control of the length $i$ as the number of sampling iterations [53].

## 3.2 Flow Models

Like autoregressive models, flow models are also belong to fully observed likelihood-based models. Flow models are generative models based on maximum likelihood estimation and their characteristic invertible neural networks. The advantages for this kind of models includes it can generate data flexibly and quickly, and it is easier to calculate the maximum likelihood estimation [54].

The principle idea of flow models is that the real data distribution is able to be mapped from the transformation function to a artificially defined simpler distribution [55]. If the such transformation function is invertible and the form of the transformation function can be obtained, then the generative model constructed by the invertible function of this transformation function and the defined simple distribution can explicitly learn the probability distribution of $p(\mathbf{x})$ like shown in II.3:

$$p(\mathbf{x}) = p(\mathbf{z}|\det(\mathbf{J}(\mathbf{x})|)\ ,\ \mathbf{z} = f(\mathbf{x}) \tag{II.3}$$

While $\mathbf{J}$ denotes the Jacobian matrix, and $\mathbf{z}$ represents the transformation function for $\mathbf{x}$. Through applying a sequence of such invertible transformation functions, like flows, the flow model can hence finally estimate the probability distribution for the target [56].

## 3.3 Latent Variable Models

Variational Autoencoder (VAE) is the most representative method as latent variable model for deep generative modeling. VAE actually is mainly inspired by variational bayesian, or say, approximate inference, and autoencoders [57]. Models with encoder-decoder architecture like autoencoders are usually with relatively strong representational ability, meanwhile the property of latent space also makes VAE be able to achieve many downstream tasks efficiently [58]. Since the complexity of the transformation for the neural networks in high-dimensional space, it is intractable to directly optimize the maximum likelihood through neural networks, while variational inference is one of the most powerful tools which is able to deal this. By introducing a new variable, as latent distribution $\mathbf{z} \sim \mathcal{N}(0, \mathrm{I})$ to variational autoencoder, the probability distribution can be described as II.4:

$$\begin{aligned} p(\mathbf{x}) &= \int p(\mathbf{z})p(\mathbf{x}|\mathbf{z})d\mathbf{z} \\ \mathbf{z} &\sim p(\mathbf{z}) \\ \mathbf{x} &\sim p(\mathbf{x}|\mathbf{z}) \end{aligned} \tag{II.4}$$

Among this equation, data $\mathbf{z}$ are sampled from $p(\mathbf{z})$, and $\mathbf{x}$ can be flexibly sampling from $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mu_\theta(\mathbf{z}), \sum_\theta(\mathbf{z}))$, where $\mu_\theta$ and $\sum_\theta$ are all neural networks. However, since the difficulty on summing all of $\mathbf{z}$ together, a tractable distribution $q_\phi(\mathbf{z}|\mathbf{x})$ parameterized by $\phi$ is introduced as a probabilistic encoder for making an approximation to $p(\mathbf{z})$. And then through techniques such as Evidence lower bound (ELBO) [59] and Reparameterization Trick [60], VAE models can therefore be effectively trained to generate data.

## 3.4 Implicit Models

The essence of Generative Adversarial Networks (GAN) is to transform the likelihood function which is intractable to solve into neural networks, and let the model to fit the likelihood function with suitable parameters through the adversarial training process by itself [61]. The internal adversarial structure of GAN can be seen as a training framework, which in principle can train to generative any type of data. By optimizing the model parameters from the adversarial behavior between two types of models it consisted, the generator and discriminator, GAN can cleverly avoiding the process of solving the likelihood function. This advantage makes the GAN highly applicable and malleable, allowing the generator and discriminator to be changed according to different needs [62].

Inside the GAN model, the mian target of its generator $G$ is to produce realistic pseudo-samples as possible so that make its discriminator $D$ hardly to distinguish the true from the false, and the goal of the discriminator $D$ is to correctly distinguish whether the data is a true sample or a pseudo-sample from the generator $G$ [63]. By introducing an latent variable $\mathbf{z}$ sampling from the probability distribution $p(\mathbf{z})$, the generator $G$ can generate target $\mathbf{x}$ through learning the generating function $g(\mathbf{z})$ from the adversarial training process with discriminator $D$, like in equation II.5:

$$\begin{aligned} &\mathbf{z} \sim p(\mathbf{z}) \\ &\mathbf{x} = g(\mathbf{z}) \end{aligned} \tag{II.5}$$

The loss function of the adversarial training process between $G$ and $D$ can be defined as a minimax game, and sometimes it's hard to optimize due to the reasons such as instability in training or failure to converge [64]. When the discriminator reaches a certain level of recognition ability but cannot correctly determine the data source generated by the generator in such implicit form, the model can be considered as learning the real data distribution.

## 3.5 Energy-based Models

Energy-based models using an alternative way that by describing an energy function $\mathbf{E}$ to make $p(\mathbf{x})$ satisfied the requirement of probability density function, as shown in equation II.6:

$$p(\mathbf{x}) = \frac{e^{-\mathbf{E}(\mathbf{x})}}{\mathbf{Z}} \ , \ \mathbf{E} = \int_{\mathcal{X}} e^{-\mathbf{E}(\mathbf{y})} dy \tag{II.6}$$

Generally, energy-based models utilize Boltzmann distribution to portray the probability distribution of data. However, the normalizing constant **Z** is usually hard to compute accurately, and therefore traditional energy-based models need to elaborately design the energy function **E** making the final normalizing constant has a closed-form expression [65], but this step also limited the capability of model's representation such that make it hard to generated complex high-dimensional data effectively [66]. Inspiring from the sampling approach through Langevin dynamics [67], modern energy-based methods can omit the calculation of the normalizing constant **Z** with the designed score function, as II.7:

$$p(\mathbf{x}) = \nabla \log p(\mathbf{x}) \tag{II.7}$$

Through the deep neural networks for approximating the score function, the performance of these models is significant at several unconditional generation tasks. For example, the Google's Imagen [68] even defeat all other models at the time and became new state of the art for text-to-image generation task.

## 4 Related Work in Automatic Songwriting

Before the advent of deep generative models which are able to generate melodies from lyrics in an end-to-end approach, early work in automatic songwriting are either based on rules or statistical techniques. Such the rule based or statistical approaches are generally required abundant prior knowledge in music when designing the algorithm, as well as lots of manipulate in the process of generation. One of these approaches is to choose rhythms based on phonemes of the input lyric by a designed a rule-based matching algorithm at first, then using constructed n-gram model that training from musical notes dataset to predict pitches for rhythms to generate melodies [69]. Another rule-based work construct a system with various modules that process different elements of melody separately. It first generated rhythms from lyrics by randomly choose notes with same number as the phonemes of the word, and then harmonies can be produced from a candidate pool through manipulating the style of music. After that, through random walk algorithm on rhythms and harmonies the melodies can hereby be generated by the system [70]. In addition, the use of statistical modeling such as machine learning techniques to generate melodies from lyrics had also attracted researchers' attention, and one of the representative works is the application of Markov Chain. Scientific Music Generator (SMUG) design a system with two Markov Chain for music generation, while one of the Markov Chain model is used to decide the pitch of the melody, and the other is for producing notes' duration [71]. Moreover, Automated LYrical SongwrIting Application (ALYSIA) propose the idea that predicting note to each words in the lyric to compose a melody by random forest [72].

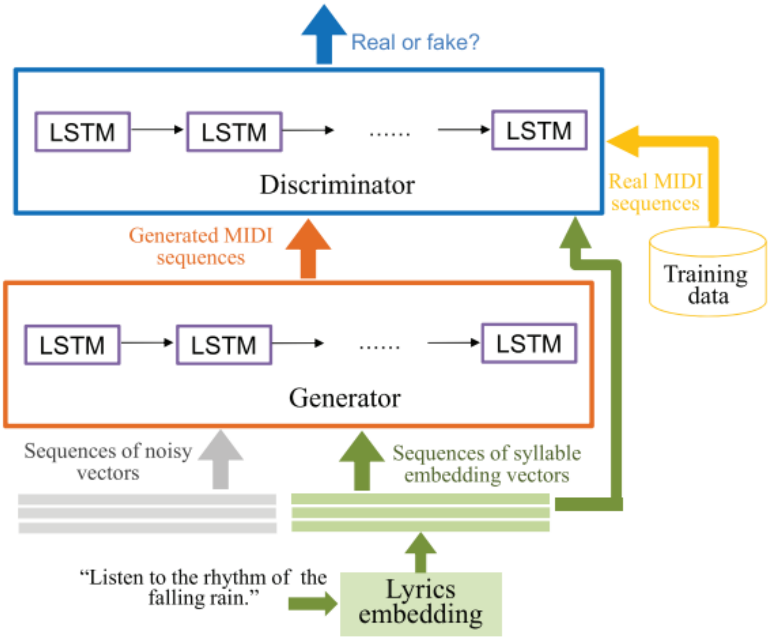

**Figure II.2** *The architecture of LSTM-GAN*

In the age of modern neural networks, the deep generative modeling techniques can not only be applied for the generation task singly, but also be combined with different approaches together. **LSTM-GAN** utilize ideas from both autoregressive modeling and implicit modeling for the challenge of generating melodies under the condition of lyrics [73]. Both of the generator and discriminator are mainly constructed by long short-term memory networks, while the LSTM in generator is used for autoregressive generating lyrics with in the generator, and LSTM in discriminator is for extract features from generated lyrics for the classify layer in order to adversarially train generator. The model structure of LSTM-GAN can be found in Figure II.2.

Inspired by Masked Sequence to Sequence Pre-training for Language Generation (MASS) [74], a encoder-decoder architecture transformer model mainly used for neural machine translation task which the decoder part is autoregressive model, **SongMASS** [74] apply such structure in both lyric-to-melody and melody-to-melody generation tasks. Regarding the problem in the lack of the paired lyric-melody data, The solution of SongMASS is to unsupervised pre-training the melodies and lyrics data through MASS architecture models of music modality and text modality respectively at first. After the unsupervised pre-training process, SongMASS interchanges two decoders that originally connected to their encoder and conducts supervised learning to achieve the ability of cross modal generation, while its architecture can find in Figure II.3:

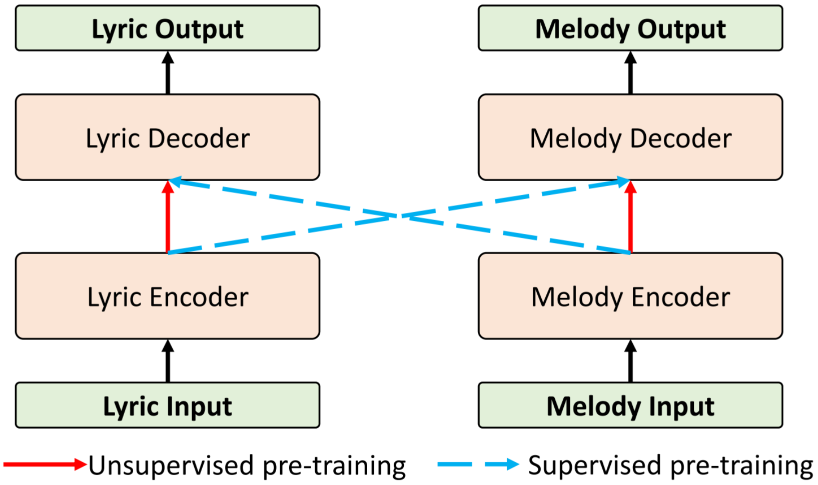

**Figure II.3** *The architecture of SongMASS*

However, none of these approaches take into account the use of knowledge that can be transfered from existing language and music pre-trained models. Rather than pre-training the cross-modal model from scratch which is required abundant relevant data and computational resources, we are more interested in maximally utilizing current pre-trained single modal models as transferring knowledge from these models can be efficient for downstream tasks. Our main goal is to explore a two-stage model combined from a pre-trained model in text modality and one in music modality, using pitch as a bridge to link the gap between lyrics and melodies, to exploit abilities in their own modality as possible for lyrics-conditioned melodies generation.

# Chapter III

# Methodology

## 1 The Two-Stage Model

In this section, we proposed a 2-stage model for the cross-modal task of lyric-to-melody generation. Although the modalities of text and music are different and they only have very weak connections, however, pitch can be considered as a bridge linking to the gap between such discrepancy.

Syllable is the smallest unit of speech in a language in which a combination of a single or compound vowel and a consonant is pronounced. In the perspective of physics, pitch is determined by the fundamental frequency of the sound, no matter the sound is come from instrument or voice [75]. Form modern Chinese phonetics, we know the pronunciation of a word, which is decided by word's phoneme in its vowel, is directed affect its pitch, simply it that the pitch of a single word can be introduced by its vowel's phoneme [76]. And several works for Singing Voice Synthesis (SVS) in Chinese had already using phonemes as input rather than characters and generated remarkable results for singing [23][24][25].

Also, as the background of music we mentioned in Chapter 1, music can be regarded as a combination of different regular beats, and beats are made up of musical notes, or say, syllables. Among each of the music note, pitch and duration are the basic units of it. Therefore, when the pitch is determined, different beats can be obtained by expanding other information, such as the duration, of the musical symbols, so that a whole section of melody can be generated through the organization of various beats as well.

Hence, by dividing the lyric-to-melody generation task as two modules, as lyric-to-pitch module and pitch-to-melody module, we can model the lyric-to-melody challenge in a simpler way, since the connection between not only the lyrics and pitches but also the pitches and melodies are much more stronger than the direct correlation of Chinese lyrics and melodies. Among these two stages, we convert lyric-to-pitch into the form of text-to-text and simplify pitch-to-melody to conditional generation task under single modality. Also, through few-shot learning and fine tuning, such modeling can maximally utilized current pre-trained deep learning models under low resources for both of the perspectives of computation and data.

The architecture of such 2-stage model is shown as in Figure III.1. Through the idea that using pitch as a bridge to link the gap between lyrics and melodies, our 2-stage model is combined from a system containing a pre-trained model in text modality for generating pitches from phonemes of the words and another pre-trained model system in music modality for generating melodies by extending musical information from pitches. Each module of this model in detail will be explained in the rest of this chapter.

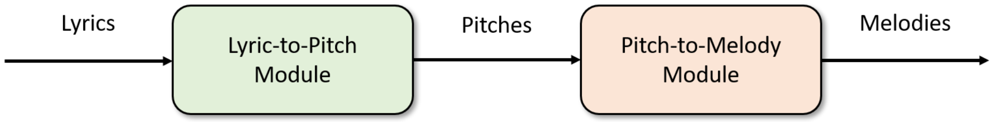

**Figure III.1** *The 2-stage modeling for Lyric-to-Melody generation*

## 2 Lyric-to-Pitch Module

The lyric-to-pitch module is used for generating pitches from given Chinese lyrics. This module is mainly composed by 2 parts, as shown in Figure III.2: the first part is a converter for converting each of the Chinese words to its according phoneme, while the second part is an attention-based encoder-decoder transformer model in text modality, through the attention mechanism it can learn the alignment of phonemes and their paired melodies. In this section, we mainly introduce the components we used for generating pitches through given lyrics for this module.

## 2.1 Lyric-to-Pitch Components

We simplified the lyric-to-pitch task with two components inside this module. After converting lyrics into their corresponding phonemes, we further reduce the task difficulty of lyric-to-pitch cross modal generation by transforming Phoneme-to-Pitch generation into format of text-to-text.

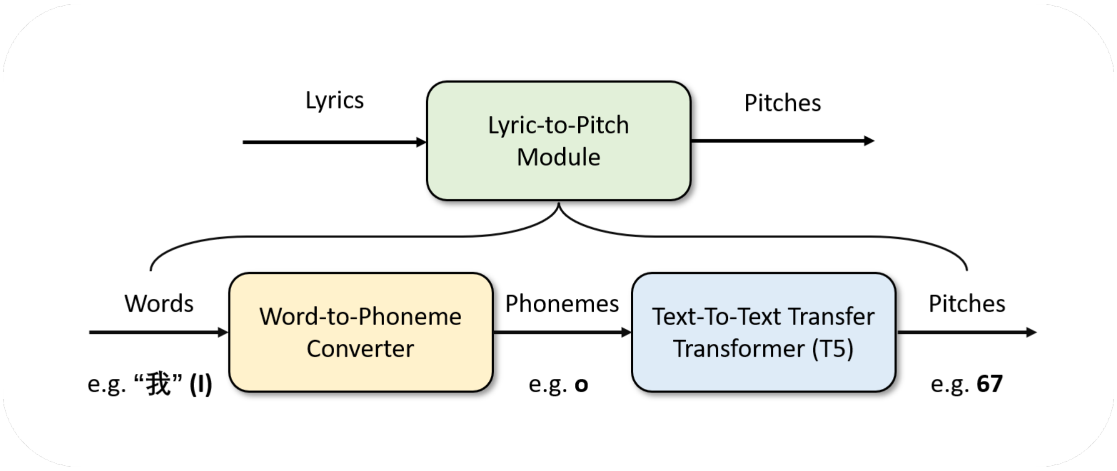

**Figure III.2** *The Lyric-to-Pitch Module*

### 2.1.1 Word-to-Phoneme Converter

Since phoneme mainly decide the pronunciation of characters, so we first transform given Chinese lyrics to their corresponding phonemes. The converter we used is **pypinyin**, which is the Python version of pīnyīn. It's a dictionary-based tool for transforming Chinese word to Pinyin, which is a romanization system that using letters as the representation for spelled sounds of Chinese, and can be used for phonetic notation, sorting and retrieval of Chinese characters [77]. Collected through different sources for single Chinese characters, phrases, and phonemes, for the heteronym pypinyin is still able to intelligently match the most correct pinyin according to the phrase. In this process, we using pypinyin to transform each Chinese characters in the lyrics to pinyin, and then only keep their phonemes inside pinyin, meanwhile maintaining the paragraph structure unchanged.

### 2.1.2 Text-To-Text Model

After convert Chinese characters to phonemes, the purpose of the another component in lyric-to-pitch module is to model the gap between phonemes and pitches, so that the model can generate aligned pitches from given phonemes of input information. Meanwhile, one thing need to mention is that in songwriting, every syllable or word in lyrics must has one unique strict melody aligned. However, some of words in lyrics may have more than one melody aligned, and this is because of the pitch transposition in music [78]: it's a special technique in singing and songwriting that indicates performer switching the note continuously, with usually two to four switches, when sing a specified syllable in the song. In addition, there is no difference between the representation of the syllable which needed to transposing pitch and others in the lyrics, but in its aligned melodies there are more than one notes while others only have one.

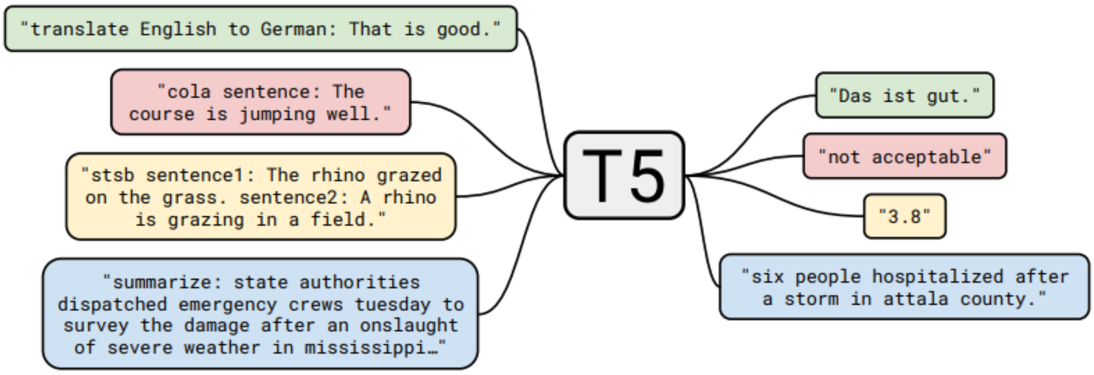
"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

**Figure III.3** *The diagram of the Text-to-Text Transfer Transformer framework*

We hence consider phoneme-to-pitch as a task in form of text-to-text, and the sequence-to-sequence (Seq2Seq) architecture model, that converting a sequence from one pattern to another without strict requirement in length, is very suitable for dealing with the generating problems mentioned above. The model we chose in this part is the **Text-to-Text Transfer Transformer**, or say, **T5**. Pre-trained through Colossal Clean Crawled Corpus (C4) which size is around 750 GB, T5 is an encoder-attention-decoder architecture transformer model in text modality, and it offers a general diagram for natural language processing (NLP) task that converts every tasks in NLP into formet of text-to-text [79], as in Figure III.3. And therefore, the T5 model is able to handle various problems in text modality as long as convert the task into suitable text inputs and outputs. Transforming the task in format of text-to-text, we hereby using phonemes as text input for T5 model and output their aligned pitches in form of text representation.

# 3 Pitch-to-Melody Module

The pitch-to-melody module receives pitches generated from lyric-to-pitch model as input and generates melodies though extending other musical elements (i.e., bar, position, duration, and tempo) from input pitches. In this section, we first introduce the attention mechanism based Transformer encoder model for symbolic music understanding we used and its encoding approach for symbolic music, and then is the technique for generating other necessary musical elements to producing a complete melody from pitches.

## 3.1 Pitch-to-Melody Model

Inspired by Roberta [80], a robustly optimized version of Bert [81], MusicBERT is also a transformer encoder that pre-trained with masked language modeling (MLM) technique. However, unlike the above-mentioned Bert-series models that were pre-training based on data in text modality, MusicBERT pre-trained on encoded symbolic music dataset Million-MIDI Dataset (MMD), which contains 1,524,557 songs with 2,075 millions of musical notes, by masking certain tokens in the input sequence of musical data and predicting them in the output [82]. By encoding the symbol music, MusicBERT effectively converts the data of the music format into a token format that can be entered by the Bert-series model.The method of this encoding is called Octuple Encoding. It converts each musical symbol into a sequence containing 8 tokens, where each token represents one of the basic musical elements in the note, and these 8 tokens including:

- **Bar** and **Position**, which reflect the positional information of this note in the piece of music, while there are 256 values in range of 0 to 255 for bar, and 128 tokens from 0 to 127 for position;
- **Instrument** indicates the type of instrument used for this note, and totally 129 tokens in Octuple Encoding for representing instruments;
- **Pitch** and **Duration** denote the pitch, which has 128 values for measuring its value, and type, which also including 128 tokens for this musical element, of the note;

- **Velocity** contains the loudness of a sound for this note, which has 32 different tokens;

- **Time Signature** and **Tempo** represent the rhythm information for the piece of music where the note is located, while denoted as a fraction, time signature involves 254 various of tokens, and the tempo as beat per minute (BPM) has 49 diverse values.

Then, through connecting the embedding tokens of 8 musical elements and concatenating them into a single vector using a linear layer, each of the Octuple tokens in the sequence represented for musical information can be transformed into the input of MusicBERT's transformer encoder. The architecture of MusicBERT can be find at Figure III.4

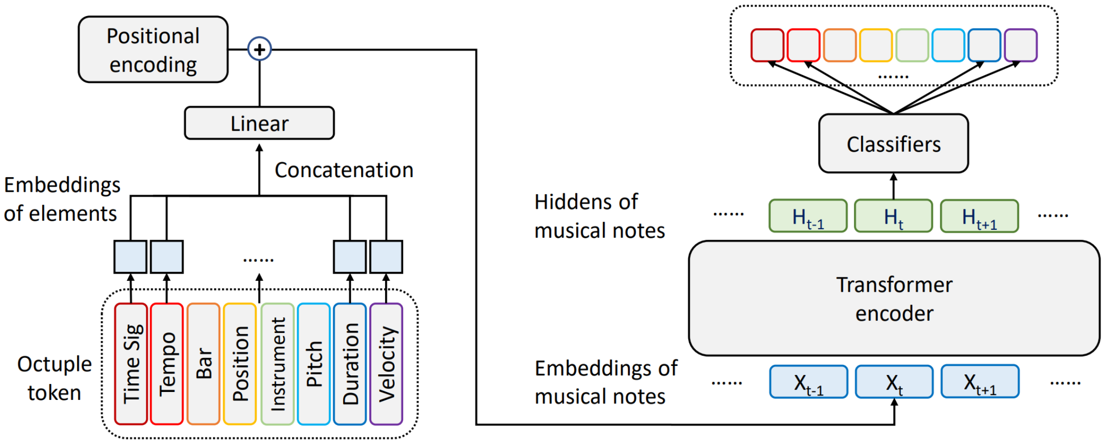

**Figure III.4** *The architecture of MusicBERT with classifier*

## 3.2 Generation as Classification

Unlike the architecture of decoder-only (i.e., GPT-3 [83]) or prefix language model (i.e., UniLM [84]) that only the previous time step information is visible for the current time step, or say, left-to-right, in the generation process, Bert-series model is bidirectional transformer encoder which is able to utilizing contextual information of the input sequence. This feature of Bert-series model makes it as generative model and enables it possible to generate fluent but also diverse output which with the similar high-quality as the traditional left-to-right generative models as well [85]. The secret of it lies in one pre-training method of Bert-series models, **Masked Language Modeling (MLM)**.

Masked language modeling can be regarded as a fill-in-the-blank task, which randomly masks some tokens among the input of text (generally for 15%), and then needs the model to predict masked words. The part that is masked can be a token that is directly randomly selected, or a continuous token that can form a whole word (Whole Word Masking [86]). Initially, masked language modeling was treated as a pre-training task for Bert-series models, which could be ignored after pre-training. As the research progressed, the researchers found that not only Bert's encoder, but also the MLM technique used for pre-training is useful for downstream tasks such as generation, text classification, and spelling error correction [85][87][88].

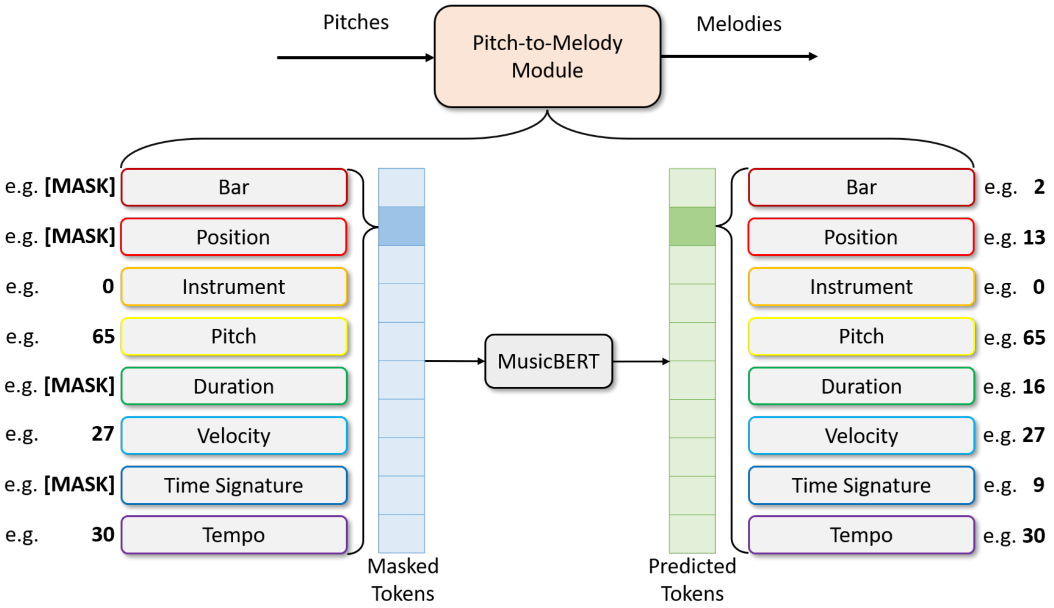

**Figure III.5** *The Pitch-to-Melody Module*

The basic idea for generating content with masked language modeling is to predict the mask token, aka '**[MASK]**', in the input sequence of tokens to the transformer encoder. Given a sequence, by inserting the mask token to the location of generation target, the bidirectional model can thereby combine contextual content of input tokens to make a prediction for these mask tokens to achieve the effect of generation. Back to the pitch-to-melody module, the length of input is already decided by generated pitches from lyric-to-pitch module, and meanwhile since instrument, velocity are weakly relevant to composing songs and time signature for music can be defined by generation style, as an input variable (i.e., $\frac{4}{4}$ for generating pop music), so these three elements in Octuple Encoding can be set by demand.

The design pattern can be found in Figure III.5. Since the number of musical notes is already determined by the length of the pitch sequence, by manipulating instrument, velocity and time signature as the value of pop music, and meanwhile setting other musical elements, which are bar, position, duration, and tempo, of the Octuple token to '[**MASK**]', the input sequence to MusicBERT can hereby obtained. Compare to traditional generation approaches with decoder models, another character of this design is that it can utilize the information contextually through bi-direction rather than only left-to-right. Although in the content part of input tokens there are 50% tokens are mask token for MusicBERT model, which is over the generally 15% normal Bert-series models used for pre-training, while some of researches already indicated that even training with 80% masked token Bert-series models can still persist most of their performance [89]. Therefore, melodies can be generated through make a classification to the mask tokens of the input sequentially and meanwhile replacing the previous mask token to its prediction result when making a classification to new mask token.

### 3.2.1 Filter for the classifier

Since there are 4 type of tokens (bar, position, duration, tempo) among Octuple Encoding are set as mask token, the candidate space for tokens in each of these four categories is needed to be constructed in order to making the prediction efficiently. Our approach is to insert a filter, as a corresponding constraints for decoding, which is integrated with domain-knowledge in music to the first layer of the model classifier that built through multilayer perceptron.

The purpose of adding such filter is to excluding invalid tokens for prediction, so that enhance both the efficiency and accuracy for training and inferencing of the MusicBERT. The filter algorithm is shown as in Algorithm 1, while $M_m$ and $N$ are represent respectively as the musical element of current mask token in current note $M$ and the previous Octuple token in the input sequence, and $logits_{MusicBERT}$ is the logits for musical element of current mask token that extract from the last layer of MusicBERT. The bar of first musical note can only start from 0, while other bars from continued notes must be either as same as previous bar or previous bar + 1, which is depend on whether previous note leaves enough positional space for current note's duration; If beginning with a new bar, then current note can located every where for the position, else the possible range is in the end of last note and the last of position of the bar; Similarly for duration, it can have all 128 possible options if it with a new bar, then the duration can be any value from 1 to max position minus current note's position, or it also need to subtract previous note's position and duration; And the value of tempo cannot be changed after it is determined by the first note.

**Algorithm 1** Filter for classifying mask tokens

```
Input: $M_m$; $N$; $logits_{MusicBERT}$
Check whether $N$ is exist, if not, $N \leftarrow -1$
if $m$ belongs to bar element then
    if $N = -1$ then
        filter all tokens except bar token at value of 0
    else
        if $N_{position} + N_{duration}$ <max position value then
            filter all tokens except $N_{bar}, N_{bar} + 1$
        else
            filter all tokens except $N_{bar} + 1$
        end if
    end if
else if $m$ belongs to position element then
    if $N = -1$ or $N_{bar} \neq M_{m-1}$ then
        filter all tokens except all of 128 position tokens
    else
        $position_{available} \leftarrow$ max position value - $(N_{position} + N_{duration})$
        filter all tokens except position tokens in range of ($position_{available}$, max position
        value)
    else if $m$ belongs to duration element then
        if $N = -1$ or $N_{bar} \neq M_{m-4}$ then
            $duration_{available} \leftarrow$ max position value - $M_{m-3}$
        else
            $duration_{available} \leftarrow$ max position value - $(N_{position} + N_{duration} + M_{m-3})$
        end if
        filter all tokens except duration tokens in range of (1, $duration_{available}$)
    end if
else if $m$ belongs to tempo element then
    if $N = -1$ then
        filter all tokens except all of 49 tempo tokens
    else
        filter all tokens except tempo token with value of $N_{tempo}$
    end if
end if
Output: filtered $logits_{MusicBERT}$
```

# Chapter IV

# Experiment

In this chapter we mainly introduce the procedure of our experiment in lyrics to melodies generation with the two-stage model. Firstly is the dataset we chose and the data processing procedure, then is the model configurations for each of the modules in our lyric-to-melody system, as well as baseline models. After that is quantitative evaluation metrics we utilized for measuring the performance for our method and other baseline models. And finally, the results of the experiment.

## 1 Dataset

We conduct the experiment on Chinese lyrics to pop music style melodies generation task for verifying the effectiveness of our designed two-stage model based on utilizing pitch as the bridge. We choose Opencpop dataset [90] as well as our already processed selected pop songs that previously purchased from music service provider through regular channels as our experiment dataset.

Opencpop is an open source dataset which initially used for Mandarin singing voice synthesis task in pop styple music. The corpus has 100 Mandarin pop music with 3756 utterances, and the sound part only contains lyric-paired melodies and without any accompaniment. One of its great advantages is that it has a very detailed annotation for the melodies and lyrics at syllable level, and this dataset contains both musical information of songs in TextGrid format, which including both lyrics and their paired pitches at word level and can be extracted to text information about lyrics, and MIDI format, which represents melodies information in the music. By matching the pitch in both the TextGrid file and the MIDI file, we can get all the note information corresponding to each word of the lyrics in the dataset.

In addition, in order to supplement dataset used for our experiment, we also select 50 pop music in Mandarin from songs our purchased. To extract the needed information for paired lyric-melody from our exist music files, we first use the Python library **audio-to-midi** convert songs in mp3 format to MIDI format, then we apply **Phonemizer** to extract human voice from accompaniment for getting lyric-paired melodies [91]. Next, we convert every words from lyrics file into phonemes with **pypinyin**, using these phonemes to match music notes in extracted melodies by **Montreal Forced Aligner**. After that, we finally get 50 Mandarin pop songs with 1803 utterances in total as the supplementary dataset to Opencpop.

Together, we have 150 songs in pop style with 5559 utterances overall. We split training, validation, and testing dataset with the ratio of 8:1:1 at song level, as 120 songs, 15 songs, and 15 songs accordingly.

## 1.1 Lyric-to-Pitch Dataset Preparation

The data features we used in our lyric-to-pitch module are phonemes and their paired pitches. We process the dataset for experimenting the model at the length of utterance level in lyrics, and specifically, there are 4452 utterances for training, 545 utterances for validation and 562 utterances for testing dataset. Table IV.1 shows two examples as data format for lyric-to-pitch module, while the *Phonemes* is the format of input data, the *Pitches* is format of target data, and ( 67 64 ) in last example means the pitch transposition for phoneme *in* in the lyric

| Original Lyrics | Phonemes | Pitches |
|---|---|---|
| 低下头俯瞰陆地上想念的眼睛 | i ia ou u an u i ang iang ian e an ing | 65 67 65 67 65 67 65 65 69 67 67 65 62 |
| 如果云层是天空的一封信 | u uo un eng i ian ong e i eng in | 69 65 67 65 65 69 65 65 69 65 ( 67 64 ) |

**Table IV.1:** An example of data format for Lyric-to-Pitch module

### 1.2 Pitch-to-Melody Dataset Preparation

In the data preparation procedure for pitch-to-melody module, we first convert all melodies data to Octuple token representation from their MIDI format. Then, for each element in set of bar, position, duration and temp in the song, we mask it and all other elements in the set behind, meanwhile using its label as the ground truth of the masked token, to let model predict the this mask token under incomplete contextual information. Through this approach, we not only predict the durations which are necessary to compose musical notes for the melody, but also modeling the structure of the melody with positional elements in beats. Table IV.2 displays an example of the data processing steps for the first note in a piece of melody. After the data pre-processing, we get 179351 amount of data can be used in the experiment of our pitch-to-melody module in total. Therefore, for the training set, the validation set and the testing set, we have 143479, 17848, and 18004 amount of data respectively.

| Label | Bar | Position | Instrument | Pitch | Duration | Velocity | Time Signature | Tempo | ... |
|---|---|---|---|---|---|---|---|---|---|
| 0 | [MASK] | [MASK] | 0 | 68 | [MASK] | 27 | 9 | [MASK] | ... |
| 28 | 0 | [MASK] | 0 | 68 | [MASK] | 27 | 9 | [MASK] | ... |
| 5 | 0 | 28 | 0 | 68 | [MASK] | 27 | 9 | [MASK] | ... |
| 30 | 0 | 28 | 0 | 68 | 5 | 27 | 9 | [MASK] | ... |

**Table IV.2:** An example of data processing procedure for Pitch-to-Melody module

## 2 Model Configurations

In this section, we mainly introduce the specific configuration setting for our designed model and baseline models for comparison in the experiment.

### 2.1 Lyric-to-Pitch Module

We use the T5-small model in the lyric-to-pitch module for generating pitches from phonemes that converted through lyrics. T5-small is a pre-trained version of T5 model, which is a natural language processing model pre-trained by Colossal Clean Crawled Corpus (C4) with size of 750 GB in language of English, French, German and Romanian. Its 6 layers encoder-decoder architecture and 60 million parameters bring it strong capability for transfer learning.

T5-small has 8 headed attention with size of 512 for encoder layers and 2048 for intermediate feed forward layer in each block. We fine tune the T5-small model on our processed phoneme-pitch dataset with prompt-based learning [92], by adding a discrete prompt *"Generate pitches from phonemes: "* in front of input phonemes sequence during the procedure for each input. And the optimizer we use for fine tuning T5-small with cross-entropy loss is Adam [93], meanwhile the learning rate 0.0001 and drop out rate 0.1 for 30 training epochs.

## 2.2 Pitch-to-Melody Module

The model we choose for pitch-to-melody Module is MusicBERT-small with our designed filter classifier. MusicBERT-small is a checkpoint of MusicBERT which was pre-trained on the *LMD-full dataset* of *Lakh MIDI dataset* [94] with size of 178561 MIDI melodies. The MusicBERT-samll is a 8 attention heads model, and it has 4 encoder layers with size of 512 for hidden size and 2048 for FFN inner hidden size. The classifier is a 3 layer multilayer perceptron with size of 1237, which is the total number of tokens in MusicBERT, for each layers. We use cross-entropy loss for fine tuning the combining model of MusicBERT and filter classifier with Adam optimizer on our processed note prediction dataset, setting the learning rate to 0.0001 and drop out rate as 0.1 from 30 training epochs. In addition, to verify the effectiveness of the filter with domain-knowledge integrated to MusicBERT, we also train another model, that without the filter in the classifier, with same configurations.

## 2.3 Baseline Models

We choose two generative models that generating symbolic musical melodies from lyrics we mentioned in Related Work section of Chapter 2 Background as our baseline models. These two models are LSTM-GAN [73] and SongMASS [95] respectively, they are all already published state-of-the-art models in recent years. Since the models we used in lyric-to-pitch module and pitch-to-melody module are both pre-trained models for transfer learning, it's inappropriate for training these baseline models from scratch when comparing the generation performances with our method. Therefore, during the situation of low resources, we choose to fine tune the pre-trained version baseline models for both of the LSTM-GAN and SongMASS, with same training epochs and dataset we applied for our model in training, validation and testing.

The LSTM-GAN model we used as the baseline using Recurrent Neural Networks for both of its generator and discriminator. The generator contains three layers as the first is fully-connected layers and the rest two are LSTM; while the first two layers in totally three layers of the discriminator are also LSTM, and its last layer is the binary classifier layer. The pre-trained version of LSTM-GAN is trained through *LMD dataset* with 7998 lyric-paired melodies, and we fine tune it with adam optimizer on cross-entropy loss using 0.0001 as learning rate.

The pre-trained model for MASS was trained through 65954 lyric-melody upaired dataset from *380000+ lyrics from MetroLyrics* and the *LMD-full dataset* and 7998 lyric-paired melodies from *LMD-matched dataset* in *Lakh MIDI Dataset*. For the SongMASS model it has 6 encoder-decoder layers with 8 attention heads, and each layer has the hidden size as 512 and 2048 for the filter size. Similar as our model, Adam optimizer is utilized for its training process with cross-entropy loss, that learning rate as 0.0001, and drop out rate is set to 0.1.

# 3 Evaluation Metrics

In this section, we introduce our evaluation metrics for generated melodies through given lyrics input. Since there is no unique standard answer for the content generation task, in addition to objective evaluation for measuring melodies between generated results and ground truth. Meanwhile, since it is difficult to measure the musicality of the generated melody by programmed objective criteria, we also invited 5 participants in the music industry to conduct subjective scores on these generated results.

## 3.1 Objective Evaluation Metrics

Our objective evaluation to the generated melodies is mainly about measuring the similarity between their musical notes and notes in ground truth melodies. Due to space of reasonable melodies in music is vast and intractable to quantify, it is obviously not appropriate to use accuracy between ground truth and generated notes to measure performance of model. We consider two metrics for the evaluation, which are similarities in **pitch distribution** and **duration distribution** among generated melodies and their ground truth [96]. Meanwhile, since pitches are generated from lyric-to-pitch module and durations are generated from Pitch-to-Melodies module, for these two evaluation metrics the similarity in pitches can reflect performance for lyric-to-pitch module, and similarity in durations can measure results of both modules but more in pitch-to-melody module.

By computing the distribution of pitches and durations for each of the musical notes among generated melodies and ground truth melodies, we can calculate their similarity through equation IV.1 and IV.2 respectively:

$$Similarity_{Pitch} = \frac{1}{N}\sum_{i=1}^{N} OA(Distribution_{Pitch}^{(i)}, Distr\hat{i}bution_{Pitch}^{(i)}) \tag{IV.1}$$

$$Similarity_{Duration} = \frac{1}{N}\sum_{i=1}^{N} OA(Distribution_{Duration}^{(i)}, Distr\hat{i}bution_{Duration}^{(i)}) \tag{IV.2}$$

Where $N$ represents the total number of tested melodies, $OA$ is the function for measuring the overlapped area among all of the musical notes in the melody averagely, and $Distribution$ and $Distr\hat{i}bution$ are the accordingly musical element (pitch, duration) distribution for the generated result and ground truth of the melody.

## 3.2 Subjective Evaluation Metrics

For the subjective evaluation to our two-stage model and baseline models' generated results, we invited five participants with professional academic background in either of theoretic music or music performance, and they all from prestigious art institute such as Royal College of Music, Royal Academy of Music and Academy of Art University.

Inspiring by previous approaches in subjective rating for evaluating the generated melodies quantitatively [97][98], we require each of the 5 participants rate the generated results in range of integer 1 to 5, where 1 means lowest score and 5 is the top evaluation, with following 3 properties of the music:

- **Fluency** measures the harmonious degree of melody. This indicator not only reflects the consistency of the rhythm style, but also reflects the rationality of the notes in the melody;
- **Diversity** indicates the variety of beats in the melody. A low value means that the song is monotonous, otherwise it means that the rhythm in the melody is rich and diverse;
- **Quality** represents the overall rating to the musicality of the generated melodies from the participant.

# 4 Results

In this section, we first compare the evaluation scores of generated results for our two-stage model, in both with and without the domain knowledge integrated filter, and baseline models. After that, we show further method analysis on generated melodies for these models, and supplement with case study.

## 4.1 Main Results

After training with 30 epochs for each of the models we test, we made statistics on the experimental results, and the main results we obtained for both in objective metrics and subjective metrics are shown in the Table IV.3.

| Model Name | Objective Metrics | | Subjective Metrics | | |
|---|---|---|---|---|---|
| | Similarity$_{\text{Pitch}}$ | Similarity$_{\text{Duration}}$ | Fluency | Diversity | Overall |
| LSTM-GAN | 38.75 | 36.94 | 2.11 | 1.25 | 1.79 |
| SongMASS | 45.20 | 53.87 | 2.48 | 2.13 | 2.32 |
| 2-stage model$_{no\ filter}$ | 70.48 | 54.13 | 2.31 | 2.61 | 2.44 |
| **2-stage model$_{with\ filter}$** | **70.52** | **58.30** | **2.51** | **3.16** | **2.71** |

**Table IV.3:** Evaluation results for our two-stage model and baselines

We can see our two-stage models, either with or without the added filter for the pitch-to-melody module, are outperform the rest of baseline models, LSTM and SongMASS, in Chinese lyric-based melody generation task for every evaluation metrics. These outcomes represent the effectiveness of our two-stage modeling approach for the cross modal generation task of lyric to melody.

Get benefit from the powerful transfer learning capabilities of the T5 model, the quality of generated pitches by our approach's lyric-to-pitch module is significantly ahead of the baseline models, which are pre-trained on 'limited' lyric data (compare with C4). Even the relative poorer one in our tested 2-stage models is still 25.28% higher than SongMASS, which with the best performance in the baseline model, in the score of pitch similarity to the ground truth.

For the experimental results in the similarity between generated duration of notes and their true label, technique of 2-stage modeling also has better performance than baseline models, and this indicated the correctness of our method that using masked modeling technique of musical modality pre-trained model to generated musical elements.

Meanwhile, same conclusions can be made from subjective metrics. From participants that we invited to evaluate generated results by these experimental models, the final subjective scores we obtained also show that the two-stage models have better performance for lyrics to melodies generation comprehensively than baselines. Our method not only achieves better results on objective indicators, but also produces more natural melodies on the subjective side. All of these results demonstrate that our modeling approach that dividing original task to two sub-tasks in lyric-to-pitch and pitch-to-melody for lyric-to-melody generation is effective.

## 4.2 Method Analysis

We then verify the effectiveness of our proposed 2-stage model as well as the classifier filter in melodies generation from both the objective and subjective perspectives. The main role of this filter, which is design from principles of music theory, is to ensure that in the process of generating melody based on the pitch sequence input, the newly generated musical notes will not overlap with the positions of the previously generated notes.

The performance in generated pitch for 2-stage model$_{no\ filter}$ and 2-stage **model**$_{with\ filter}$ are similar is because both of them share a same lyric-to-pitch module which is mainly composed by a fine tuned T5 model, and the tiny difference between their generated results is due to the randomness from the beam search decoding techniques we used for T5. Such results prove the correctness of our idea in modeling pitches. By simplify the lyrics to melodies task through substituting lyrics to phonemes and melodies to pitches at stage one of our model, and converting it to the format of text-to-text so that the pre-trained language model can easily be applied here, we hence reduce the complexity for generating melodies form lyrics a lot.

However, the outcome of duration similarity for 2-stage model, that without designed filter in the classifier, is more closed to one of the better baseline model SongMASS rather than the model performs best in this metric, the 2-stage model that integrated with domain knowledge block. Our model with such filter is actually 4.17% better than the vanilla one on duration generation, and such result indicate the effectiveness of inserting principles of music theory to neural melody generation. Through using domain knowledge to filter out most of the parts that are not relevant to the current task in the process of melody generation, the model with filter can focus more on the pattern of current generation method in the learning process, to produce better results.

Meanwhile, from participants that we invited to evaluate generated results by these experimental models, the final subjective scores we obtained also show that the two-stage model which utilized domain knowledge in music has the best performance for lyrics to melodies generation comprehensively. Especially in terms of the diversity of generated musical notes, this model is greatly outperforms than others. By optimizing the position of musical notes in generating beats, the filter we designed can control the together fine tuned pre-trained model MusicBERT to generate melodies with better structure.

These results prove the efficiency of the 2-stage modeling approach we proposed for utilizing limited lyric-melody paired data, as well as the advantage in use of the designed filter for enhancing melody generation under low resource.

## 4.3 Case Study

To further demonstrate our discovery on the effectiveness of the two-stage modeling approach with a integrated block, we show melody samples that generated from various techniques introduced in our experiment and their corresponding ground truth in form of music sheet, as in Figure IV.1.

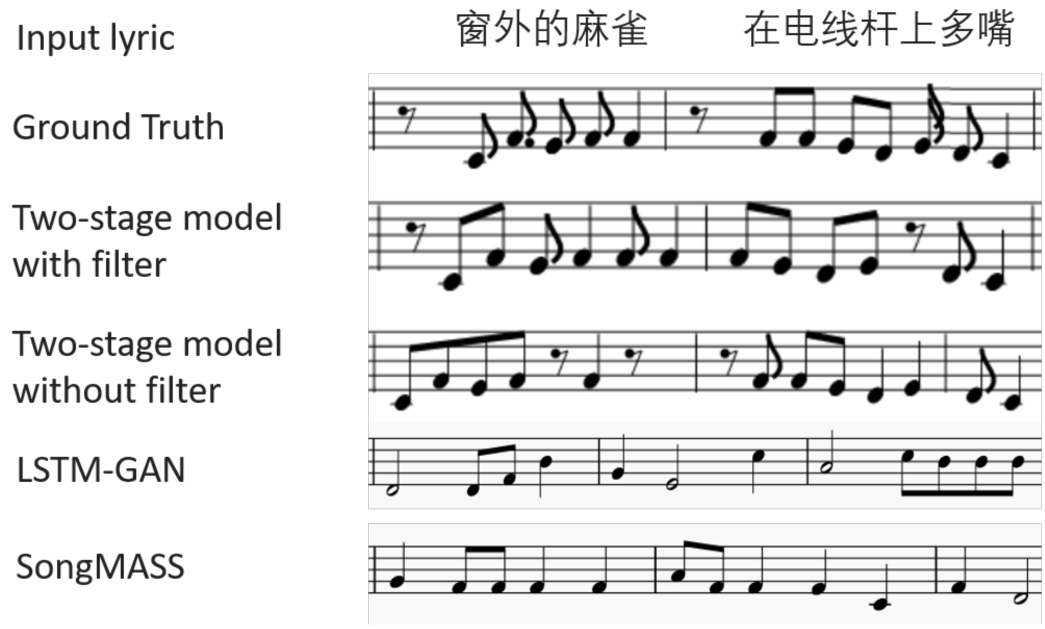
Input lyric
窗外的麻雀 在电线杆上多嘴
Ground Truth
Two-stage model with filter
Two-stage model without filter
LSTM-GAN
SongMASS

**Figure IV.1** *Case study in generated melodies*

Obviously, we can observe that the melody generated from our two-stage model with filter generate in this example is more natural and harmonious compare with other baseline models and the two-stage model without the designed filter in pitch-to-melody module. Meanwhile, our filter does not have too much restrictions on the on the freedom of melody generation, and it still retains the flexibility of the model during the process of lyric-based automatic songwriting.

# Chapter V

# Conclusions

## 1 Summary

In this dissertation, we proposed and implemented a two-stage modeling approach, using pitch as a bridge to link the gap between lyric and its paired melody, for generating melodies based on lyrics in the task of automatic songwriting. In addition, both of the subjective evaluation and objective evaluation from the experiment in this cross modal challenge indicate our method is able to be competent in generating harmonious and natural melodies from given lyrics and better than previous related works of lyrics conditional melodies generation. Meanwhile, our contributions in this thesis are multiple:

1. From previous works in singing voice synthesis (SVS) systems that indicated converting texts into phonemes of Chinese lyrics as the input can effectively generate music in audio modality [23][24][25], we applied the similar idea of using phonemes denote lyrics in the task of generating melodies from lyrics. For generating music in form of symbolic representation based on lyrics, our experiment results have also demonstrated the effectiveness in using phonemes to substitute texts in lyrics.

2. Inspired from the framework of hierarchical reinforcement learning, as transforming a complex task into several hierarchical sub-tasks that are easier to solve [99], we divide the lyric-to-melody generation challenge into two stages, in lyric-to-pitch and pitch-to-melody, with pitch as the bridge

to link the modality gap between lyrics and melodies. Among these two stages, we convert the sub-task lyric-to-pitch into form of text-to-text, make it can be easily solved by sequence to sequence approaches. Besides, we simplify the sub-task pitch-to-melody to single modal conditional generation task which can hereby using one of exist music model to handle with. The objective evaluation metrics in the experiment used to reflect these two tasks also show the advantages of our model, and these indicate the validity of the idea in two-stage modeling for lyric-to-melody generation

3. Meanwhile, our two-stage modeling approach is flexible and available in knowledge transference. Through utilizing pre-trained text modality model in lyric-to-pitch module and pre-trained music modality model in pitch-to-melody module, we hereby can exploit their transfer ability for learning with limited lyric-melody paired data on the cross modal generation task data efficiently. In the case of small samples of only 120 songs, our model has achieved the results in the test better than previous related works.

4. Besides, we show the effectiveness of introducing prior knowledge into the neural melody generation system. By inserting our filter designed based on music theory to the pitch-to-melody module, our model can hence pay more attention to feasible candidate spaces in the process of learning to generate melodies, thus improving the learning efficiency and generation effect

In summary, we demonstrate the efficiency and high performance of our two-stage modeling approach, which uses pitch as a bridge to connect the gap of modality difference between lyric and melody, for the lyric-to-melody generation task of cross modal automatic songwriting.

## 2 Future Recommendations

More than lyric-to-melody generation, the framework of our designed two-stage modeling approach can also have huge potential in other areas of automatic songwriting.

The models we used in both of the lyric-to-pitch module and pitch-to-melody module are not only flexible to deploy but also capability for doing multitasks. For example, through fine tuning with prompts, theoretically the text-to-text model T5 we used in stage one can also be used to do other NLP tasks related to songwriting such as lyrics generation and lyrics adaptation but still retaining the ability in generating pitches. In addition, the MusicBERT model we used for stage two of our melody generation system can also be fine tuned or prompt tuned to work with several downstream tasks related to automatic music creation, such as musical style transfer and accompaniment generation. If we choose the suitable prompt tuning approaches, we might able to let MusicBERT achieve all of these downstream tasks, including our original pitch-to-melody generation, through only corresponding prompt batches so that it don't need to store model in different fine tuned versions for their corresponding downstream tasks [100]. Combining these potential abilities together, we can even extend our current two-stage model into a framework that able to create a song from zero.

Besides, current speed efficiency of generating melodies still have space to be improved. Our present method is using a classifier to predict the foremost mask token in the input sequence of pitch-to-melody module, and after one is classified we will replace the mask token to the token represents predicted value in the sequence and input the new sequence to the MusicBERT model to predict next mask token. Since we need to repeat such process until all mask tokens have been predicted, the speed for generating melody of a song can take more than one minute. Inspired by previous work combing two BERT model as an encoder-decoder model for sequence to sequence task [101], it would be interesting to explore the performance and efficiency for generating melodies with encoder-decoder model that combined from two MusicBERT, which we used in our model's pitch-to-melody module, for generating melodies from pitches.

# References

1. Loy, G. in *Current directions in computer music research* 291–396 (1989).
2. Hedges, S. A. Dice music in the eighteenth century. *Music & Letters* **59,** 180–187 (1978).
3. Davis, A. R. *The Penguin book of Chinese verse* (Harmondsworth, Middlesex, Penguin Books, 1962).
4. Chang, K.-i. S. & Owen, S. *The Cambridge history of Chinese literature* (Cambridge University Press, 2010).
5. Idema, W., Haft, L. & Haft, L. L. *A guide to Chinese literature* **74** (University of Michigan Press, 1997).
6. Collier, W. F. *A History of English Literature in a Series of Biographical Sketches...* (Nelson, 1882).
7. Southworth, J. The English Medieval Minstrel (Woodbridge. *Suffolk, eng* (1989).
8. Fatoni, N. R., Santosa, R. & Djatmika, D. MOOD System on Supporter Chant in English Premier League: A Systemic Functional Linguistic Study (2020).
9. FATONI, N. R., SANTOSA, R., *et al.* Analysis of Textual Meaning on Lyrics of Supporter's Chant to Support Football Players in English Premier League. *PAROLE: Journal of Linguistics and Education* **10,** 146–155 (2020).
10. Perricone, J. *Melody in songwriting: tools and techniques for writing hit songs* (Hal Leonard Corporation, 2000).

11. Robb, S. L. Techniques in song writing: Restoring emotional and physical well being in adolescents who have been traumatically injured. *Music Therapy Perspectives* **14,** 30–37 (1996).

12. Müller, M. *Fundamentals of Music Processing: Using Python and Jupyter Notebooks* (Springer Nature, 2021).

13. Müller, M. *Fundamentals of music processing: Audio, analysis, algorithms, applications* (Springer, 2015).

14. Dannenberg, R. B. & Hu, N. Pattern discovery techniques for music audio. *Journal of New Music Research* **32,** 153–163 (2003).

15. Moog, R. A. MIDI: musical instrument digital interface. *Journal of the Audio Engineering Society* **34,** 394–404 (1986).

16. Kim, J., Urbano, J., Liem, C. & Hanjalic, A. One deep music representation to rule them all? A comparative analysis of different representation learning strategies. *Neural Computing and Applications* **32,** 1067–1093 (2020).

17. Xue, L. *et al.* DeepRapper: Neural rap generation with rhyme and rhythm modeling. *arXiv preprint arXiv:2107.01875* (2021).

18. Wiggins, G., Miranda, E., Smaill, A. & Harris, M. A framework for the evaluation of music representation systems. *Computer Music Journal* **17,** 31–42 (1993).

19. Lin, J. *et al.* M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823* (2021).

20. Liebowitz, S. J. & Watt, R. How to best ensure remuneration for creators in the market for music? Copyright and its alternatives. *Journal of Economic Surveys* **20,** 513–545 (2006).

21. Hennequin, R., Khlif, A., Voituret, F. & Moussallam, M. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software* **5,** 2154 (2020).

22. Ju, Z. *et al.* TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method. *arXiv preprint arXiv:2109.09617* (2021).

23. Liu, J., Li, C., Ren, Y., Chen, F. & Zhao, Z. *Diffsinger: Singing voice synthesis via shallow diffusion mechanism* in *Proceedings of the AAAI Conference on Artificial Intelligence* **36** (2022), 11020–11028.

24. Zhang, Z., Zheng, Y., Li, X. & Lu, L. WeSinger: Data-augmented Singing Voice Synthesis with Auxiliary Losses. *arXiv preprint arXiv:2203.10750* (2022).

25. Liu, J., Li, C., Ren, Y., Zhu, Z. & Zhao, Z. Learning the Beauty in Songs: Neural Singing Voice Beautifier. *arXiv preprint arXiv:2202.13277* (2022).

26. Deng, Y. *et al.* A multimodal deep learning framework for predicting drug–drug interaction events. *Bioinformatics* **36,** 4316–4322 (2020).

27. Palmirotta, R. *et al.* Liquid biopsy of cancer: a multimodal diagnostic tool in clinical oncology. *Therapeutic advances in medical oncology* **10,** 1758835918794630 (2018).

28. Cui, H. *et al.* *Multimodal trajectory predictions for autonomous driving using deep convolutional networks* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 2090–2096.

29. Arapakis, I. *et al.* *Integrating facial expressions into user profiling for the improvement of a multimodal recommender system* in *2009 IEEE International Conference on Multimedia and Expo* (2009), 1440–1443.

30. Kennedy, L., Chang, S.-F. & Natsev, A. Query-adaptive fusion for multimodal search. *Proceedings of the IEEE* **96,** 567–588 (2008).

31. Baltrušaitis, T., Ahuja, C. & Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* **41,** 423–443 (2018).

32. Morency, L.-P., Liang, P. P. & Zadeh, A. *Tutorial on multimodal machine learning* in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts* (2022), 33–38.

33. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

34. Palaskar, S., Libovick, J., Gella, S. & Metze, F. Multimodal abstractive summarization for how2 videos. *arXiv preprint arXiv:1906.07901* (2019).

35. Bregler, C., Covell, M. & Slaney, M. *Video rewrite: Driving visual speech with audio* in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), 353–360.

36. Yagcioglu, S., Erdem, E., Erdem, A. & Cakıcı, R. *A distributed representation based query expansion approach for image captioning* in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (2015), 106–111.

37. Mason, R. & Charniak, E. *Nonparametric method for data-driven image captioning* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2014), 592–598.

38. Ordonez, V., Kulkarni, G. & Berg, T. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems* **24** (2011).

39. Karpathy, A., Joulin, A. & Fei-Fei, L. F. Deep fragment embeddings for bidirectional image sentence mapping. *Advances in neural information processing systems* **27** (2014).

40. Jiang, X. *et al.* The classification of multi-modal data with hidden conditional random field. *Pattern Recognition Letters* **51,** 63–69 (2015).

41. Murphy, K. P. *Probabilistic machine learning: an introduction* (MIT press, 2022).

42. Theis, L., Oord, A. v. d. & Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).

43. Salakhutdinov, R. Learning deep generative models. *Annual Review of Statistics and Its Application* **2,** 361–385 (2015).

44. Liu, C., Wang, C., Sun, F. & Rui, Y. *Image2Text: a multimodal image captioner* in *Proceedings of the 24th ACM international conference on Multimedia* (2016), 746–748.

45. Huang, X., Liu, M.-Y., Belongie, S. & Kautz, J. *Multimodal unsupervised image-to-image translation* in *Proceedings of the European conference on computer vision (ECCV)* (2018), 172–189.

46. Radford, A. *et al. Learning transferable visual models from natural language supervision* in *International Conference on Machine Learning* (2021), 8748–8763.

47. Akbari, H. *et al.* Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems* **34,** 24206–24221 (2021).

48. Sonia, S., Kumar, P. & Saha, A. *Automatic Question-Answer Generation from Video Lecture using Neural Machine Translation* in *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)* (2021), 661–665.

49. Xu, J., Li, H. & Zhou, S. An overview of deep generative models. *IETE Technical Review* **32,** 131–139 (2015).

50. Tomczak, J. M. in *Deep Generative Modeling* 1–12 (Springer, 2022).

51. Woodward, W. A., Gray, H. L. & Elliott, A. C. *Applied time series analysis with R* (CRC press, 2017).

52. Gregor, K., Danihelka, I., Mnih, A., Blundell, C. & Wierstra, D. *Deep autoregressive networks* in *International Conference on Machine Learning* (2014), 1242–1250.

53. Thickstun, J. *Leveraging Generative Models for Music and Signal Processing* PhD thesis (University of Washington, 2021).

54. Ho, J., Chen, X., Srinivas, A., Duan, Y. & Abbeel, P. *Flow++: Improving flow-based generative models with variational dequantization and architecture design* in *International Conference on Machine Learning* (2019), 2722–2730.

55. Albergo, M. S., Kanwar, G. & Shanahan, P. E. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Physical Review D* **100,** 034515 (2019).

56. Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* **31** (2018).

57. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

58. Gal, Y. & Ghahramani, Z. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning* in *international conference on machine learning* (2016), 1050–1059.

59. Yang, X. Understanding the variational lower bound. *variational lower bound, ELBO, hard attention* **13,** 1–4 (2017).

60. Kingma, D. P., Salimans, T. & Welling, M. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems* **28** (2015).

61. Wang, K. *et al.* Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica* **4,** 588–598 (2017).

62. Metz, L., Poole, B., Pfau, D. & Sohl-Dickstein, J. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163* (2016).

63. Goodfellow, I. *et al.* Generative adversarial nets. *Advances in neural information processing systems* **27** (2014).

64. Creswell, A. *et al.* Generative adversarial networks: An overview. *IEEE signal processing magazine* **35,** 53–65 (2018).

65. Ngiam, J., Chen, Z., Koh, P. W. & Ng, A. Y. *Learning deep energy models* in *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), 1105–1112.

66. Li, Z., Chen, Y. & Sommer, F. T. Annealed Denoising score matching: learning Energy based model in high-dimensional spaces (2019).

67. Song, Y. & Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems* **32** (2019).

68. Saharia, C. *et al.* Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

69. Monteith, K., Martinez, T. R. & Ventura, D. *Automatic Generation of Melodic Accompaniments for Lyrics.* in *ICCC* (2012), 87–94.

70. Toivanen, J., Toivonen, H. & Valitutti, A. *Automatical composition of lyrical songs* in *The Fourth International Conference on Computational Creativity* (2013).

71. Scirea, M., Barros, G. A., Shaker, N. & Togelius, J. *SMUG: Scientific Music Generator.* in *ICCC* (2015), 204–211.

72. Ackerman, M. & Loker, D. *Algorithmic songwriting with alysia* in *International conference on evolutionary and biologically inspired music and art* (2017), 1–16.

73. Yu, Y., Srivastava, A. & Canales, S. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **17,** 1–20 (2021).

74. Song, K., Tan, X., Qin, T., Lu, J. & Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450* (2019).

75. Moore, B. C. in *Human psychophysics* 56–115 (Springer, 1993).

76. Wang, W. S. & Sun, C. *The Oxford handbook of Chinese linguistics* (Oxford University Press, 2015).

77. Coblin, W. S. in *A Handbook of 'Phags-Pa Chinese* 177–212 (University of Hawaii Press, 2006).

78. Huang, C.-F., Hong, W.-G. & Li, M.-H. *A Research of Automatic Composition and Singing Voice Synthesis System for Taiwanese Popular Songs* in *ICMC* (2014).

79. Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21,** 1–67 (2020).

80. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

81. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

82. Zeng, M. *et al.* Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630* (2021).

83. Brown, T. *et al.* Language models are few-shot learners. *Advances in neural information processing systems* **33,** 1877–1901 (2020).

84. Dong, L. *et al.* Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems* **32** (2019).

85. Wang, A. & Cho, K. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094* (2019).

86. Cui, Y., Che, W., Liu, T., Qin, B. & Yang, Z. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **29,** 3504–3514 (2021).

87. Garg, S. & Ramakrishnan, G. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970* (2020).

88. Zhang, S., Huang, H., Liu, J. & Li, H. Spelling error correction with soft-masked BERT. *arXiv preprint arXiv:2005.07421* (2020).

89. Wettig, A., Gao, T., Zhong, Z. & Chen, D. Should You Mask 15% in Masked Language Modeling? *arXiv preprint arXiv:2202.08005* (2022).

90. Wang, Y. *et al.* Opencpop: A High-Quality Open Source Chinese Popular Song Corpus for Singing Voice Synthesis. *arXiv preprint arXiv:2201.07429* (2022).

91. Bernard, M. & Titeux, H. Phonemizer: Text to Phones Transcription for Multiple Languages in Python. *Journal of Open Source Software* **6,** 3958. `https://doi.org/10.21105/joss.03958` (2021).

92. Liu, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586* (2021).

93. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

94. Raffel, C. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching* (Columbia University, 2016).

95. Sheng, Z. *et al. Songmass: Automatic song writing with pre-training and alignment constraint* in *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (2021), 13798–13805.

96. Zhang, C. *et al.* ReLyMe: Improving Lyric-to-Melody Generation by Incorporating Lyric-Melody Relationships. *arXiv preprint arXiv:2207.05688* (2022).

97. Zhu, H. *et al. Xiaoice band: A melody and arrangement generation framework for pop music* in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), 2837–2846.

98. Watanabe, K. *et al. A melody-conditioned lyrics language model* in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), 163–172.

99. Pateria, S., Subagdja, B., Tan, A.-h. & Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)* **54,** 1–35 (2021).

100. Lester, B., Al-Rfou, R. & Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

101. Guo, J. *et al.* Incorporating bert into parallel sequence decoding with adapters. *Advances in Neural Information Processing Systems* **33,** 10843–10854 (2020).