



Universiteit
Leiden

Master Computer Science

Multi-agent Symbolic Music Composition
System Based on Large Language Model

Name: Peiwen Xing
Student ID: s3838501
Date: 26/08/2025

Specialisation: Data Science: Computer Science

1st supervisor: Aske Plaat
2nd supervisor: Niki van Stein

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

With the development of artificial intelligence-generated content (AIGC) technology, AI music composition has gradually evolved from simple melody generation to complex music structure composition. However, existing tools still have limitations such as restricted generation duration, insufficient professionalism, and poor controllability. This thesis aims to utilize the semantic understanding and generation capabilities of large language models (LLMs), combined with multi-agent collaboration technology, to design and implement an efficient multi-agent symbolic music composition system, **CoComposer**, to improve the music composition quality of LLMs and lower the creative threshold for the general public.

The system constructs five agents with clear division of labor based on the Auto-Gen framework, realizes the collaborative composition process driven by agent role-playing prompts, uses the ABC notation as an intermediate carrier, and completes the full-process automation from natural language input to multi-track music output.

Experimental comparisons show that CoComposer comprehensively outperforms the existing ComposerX [1] in terms of music quality; it has more advantages in production complexity compared to a single-agent system; among different LLMs, GPT-4o has the best overall performance; although it is inferior to the dedicated music generation model MusicFX in terms of music quality, the symbolic generation path has stronger interpretability and editability.

Research shows that the multi-agent approach can effectively improve the music quality created by LLMs. At the same time, this research provides a practical reference for the design of multi-agent symbolic music composition systems based on LLMs and offers a further exploration direction for the application of the multi-agent approach in the field of AI music generation.

Keywords: LLM; multi-agent; music generation; CoComposer; ABC notation; AIGC

Contents

1	Introduction	6
1.1	Research Questions	7
1.2	Thesis Organizational Structure	8
2	Related Work and Background	9
2.1	Related Work	9
2.2	Overview of LLM	11
2.2.1	Transformer Architecture and Pretraining Mechanism	11
2.2.2	Role-Playing Language Agents	12
2.3	Overview of Multi-Agent Systems	12
2.3.1	AutoGen	12
2.4	Basic Knowledge of Music Composition	13
2.5	ABC Notation	13
2.6	Chapter Summary	14
3	System Design and Implementation	15
3.1	Limitation Analysis of ComposerX	15
3.1.1	Overview of ComposerX	15
3.1.2	Limitations of ComposerX	16
3.2	CoComposer System Design	17
3.2.1	Agent Creation Process	18
3.2.2	Agent Prompts Design	19
3.2.3	Improvements of CoComposer Compared with ComposerX	22
3.3	Chapter Summary	23
4	Experimental Design and Result Analysis	24
4.1	Experimental Objectives and Research Hypotheses	24
4.1.1	Experimental Objectives	24
4.1.2	Research Hypotheses	24
4.2	Experimental Setup	24
4.2.1	Single-Agent System Design	24
4.2.2	Comparison Objects	25
4.2.3	User Input Prompt	26
4.2.4	Evaluation Metrics	26
4.2.5	Experimental Environment	27
4.3	Experimental Process	27
4.3.1	Sample Generation	27

4.3.2	Audio Evaluation	27
4.4	Experimental Results and Analysis	28
4.4.1	Experiment 1: Performance Comparison between CoComposer and ComposerX	29
4.4.2	Experiment 2: Performance Comparison between Multi-Agent System and Single-Agent System	29
4.4.3	Experiment 3: Impact of Different LLMs on CoComposer Performance	30
4.4.4	Experiment 4: Performance Comparison between CoComposer and MusicFX	31
4.5	Research Question Answers	32
4.6	Chapter Summary	33
5	Conclusion and Discussion	34
5.1	Summary of the Thesis Work	34
5.2	System Advantages	34
5.3	Analysis of System Limitations	35
5.3.1	Inherent Limitations of Symbolic Generation Path and Model Dependence	35
5.3.2	The Boundary between Subjective Creativity and AI Generation	35
5.4	Future Research Directions	36
5.4.1	Iterative Generation Mechanism Integrating User Real-time Feedback	36
5.4.2	Model Fine-tuning Based on Music Data	36
	Reference	37
	Acknowledgements	40
A	Appendix	41

List of Figures

1 Agent Communication Pattern of ComposerX 16

List of Tables

1	Names and main responsibilities of agents in the CoComposer system	18
2	Leader Agent prompt	19
3	Melody Agent prompt	20
4	Accompaniment Agent prompt	21
5	Revision Agent prompt	22
6	Review Agent prompt	22
7	Agent prompt for single-agent system	25
8	Example user input prompt	26
9	Software environment	27
10	Statistics of sample generation quantity by music composition systems	27
11	Summary table of the results of four experiments	28
12	Twenty user input prompts	41

1 Introduction

In the late 1950s, the world's first computer-generated music was born, just a short time after the invention of the first computer. As the first musical score composed by a computer, the Illiac Suite is an early example of algorithmic music composition [2]. It uses a stochastic model (Markov chain) for generation and filters the generated materials that meet the expected characteristics through rules [3].

With the rapid development of technology, more than half a century later, artificial intelligence (AI) can now create complex musical structures. In 2019, "Symphonic Variations on My People, My Country", an AI-created work launched by the Artificial Intelligence Research Institute of Ping An of China, premiered at the Shenzhen Concert Hall. Through an automatic variation model, the work varies the melody of the song "My People, My Country" and incorporates elements of other classic songs during the variation. Its artistic perfection makes it hard for non-professional audiences to tell the difference between human and AI creation. This event marks a breakthrough of AI in the field of music creation: from generating simple melody fragments in the early stage to formally entering the stage of creating complex musical structures, promoting the potential of AI in the creative field from technical experiments to the public eye. Behind this phenomenon is the breakthrough progress of AIGC technology: when machines can understand and reproduce the logic of human artistic expression, the traditional paradigm of music creation is facing reconstruction.

Traditional music creation has long been restricted by multiple barriers. Creators need to undergo years of professional training, master systematic knowledge such as music theory, performance, and harmony arrangement. At the same time, they rely on the accumulation of inspiration, the precipitation of experience, and even professional equipment and production resources. For the vast number of music lovers, even if they have rich creative inspirations, they often find it difficult to transform their ideas into complete works due to high skill thresholds, high learning costs, or complex tool use, resulting in a large number of potential music expression needs being suppressed.

Modern AI technology is breaking down these barriers. The LLM revolution triggered by ChatGPT [4] in 2022 has brought a qualitative change to music creation. The powerful semantic understanding and generation capabilities of language models are deeply integrated with music generation technology, significantly reducing the threshold for music creation. Take MusicLM [5] as an example. It can transform text descriptions into high-quality, long-duration, and stylistically coherent music. Users only need to input text descriptions, and the model can generate music segments that match the descriptions. This text-driven generation method does not require users to master complex music theory knowledge or performance skills, enabling 'low-threshold creation' to evolve from a concept to a practical reality.

However, existing AI music tools still have obvious limitations. First, the generation duration

is limited. For example, Google’s MusicFX can only output 30-second music. Second, the input length of the prompt words is limited. Mainstream platforms such as Suno, Udio, and Mureka do not support the input of long prompt words. Third, the professionalism is insufficient. For professional singers or music producers, the music generated by these models still has a rather “toy-like” nature. Fourth, the controllability is poor. There are deficiencies in the detailed editing of music, and it is difficult to accurately achieve the refined adjustments and creative intentions of professionals for music. For example, it is impossible to carefully control the volume, rhythm, etc. of each note and each musical instrument as precisely as professional production software. Therefore, how to utilize the technical advantages of LLM to build a more intelligent music creation system that better meets the needs of the public has become an important topic in the current AI music field.

Based on the above background, this thesis designs and implements a multi-agent symbolic music composition system, CoComposer ¹, based on LLM. Through the collaborative division of labor among multiple agents, the system can achieve the full-process automation from users’ natural language input to multi-track music output. This system is targeted at ordinary users who have not received professional music training, aiming to completely lower the technical threshold of music composition and unleash the music creative potential of the public. It is hoped that the research and promotion of this system can promote the innovative development of the AI music generation field, stimulate the public’s enthusiasm for music composition, and provide reference for future related work.

1.1 Research Questions

This thesis intends to rely on the technical advantages of LLM. Through multi-agent technology and agent prompt design, it fully explores the potential of LLM in the field of music creation. Furthermore, it designs and implements a multi-agent music composition system based on LLM.

The main research question of this thesis is: Can the use of a multi-agent approach improve the quality of music created by an LLM?

The specific research questions of this thesis are as follows:

RQ1: How does the music quality created by CoComposer compare to that of the existing ComposerX?

RQ2: Can the music created by a multi-agent system exceed the quality of that created by a single-agent system?

¹<https://github.com/PhotonCombiner/CoComposer>

RQ3: What are the differences in the quality of music created when using different LLMs as the CoComposer agent model?

RQ4: Can the music created by CoComposer reach a quality level comparable to that of a specialized music-generation model (MusicFX)?

1.2 Thesis Organizational Structure

The content of this thesis will be presented in the following structure:

Chapter 1 expounds on the background and significance of the research questions in this thesis, clearly puts forward the research questions to be explored, and briefly describes the overall organizational structure of the thesis.

Chapter 2 introduces the related work of AI music generation and the relevant technologies and theoretical foundations involved in the system development, including the related concepts of LLM and multi-agent system, the basic elements of music composition, and the ABC notation, providing theoretical support for the subsequent system design.

Chapter 3 elaborates on the system design of CoComposer, including an analysis of the limitations of the existing ComposerX, the design and workflow of this system, and the improvements of this system compared to ComposerX.

Chapter 4 introduces the experimental design and analyzes the results, answering the research questions proposed in this thesis through a series of experiments.

Chapter 5 summarizes the research work of the whole text, then discusses the advantages and limitations of the system, and puts forward the future research directions.

2 Related Work and Background

2.1 Related Work

In recent years, with the rise of deep generative methods, significant progress has been made in the field of audio generation [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. Against the backdrop of the rapid development of generative artificial intelligence, the fields of audio and music creation has also witnessed a new technological revolution[5, 7, 19, 20, 21, 22, 23, 24, 25, 26, 27, 1]. From text-to-music generation to cross-modal audio understanding, a series of innovative systems driven by large models are constantly expanding the boundaries of sound expression. These tools not only demonstrate the cutting-edge capabilities of AI in music understanding and generation but also provide more efficient and diverse solutions for multiple application scenarios such as music production, education, games, and advertising, indicating that a new era of "sound as language, music as dialogue" has quietly arrived. The following is an introduction to representative AI music generation models and platforms launched in recent years:

In January 2023, Google launched MusicLM [5] (later upgraded to MusicFX [19]), and in June of the same year, Meta released MusicGen [7]. These models adopt the MuLan/Clap cross-modal and autoregressive Transformer models and can generate music with a frequency of 24kHz and a duration of up to 5 minutes based on text prompts. They support specifying parameters such as music genre, instruments, and emotions, and their works are mainly instrumental music, basically without vocals.

The Suno AI [20][21] launched in December 2023 is a generative AI tool focused on generating realistic songs with vocals and instruments through text prompts. It uses advanced deep learning and natural language processing technologies, supports multiple languages such as Chinese, English, and Japanese, as well as various music styles. Its core functions include AI composition, lyric generation and singing, and instrumental music creation. Users can adjust parameters such as speed and melody to optimize their works. This tool has a wide range of application scenarios and is suitable for fields such as personal creation, commercial production, and music education. It has also partnered with Microsoft to be integrated into Microsoft Copilot, and users can easily use it through the Discord channel or the web version.

In February 2024, the research team of Skywork AI PTE. LTD. and the Hong Kong University of Science and Technology jointly launched the open-source LLM ChatMusician [22]. By continuously pre-training and fine-tuning the LLaMA2 model and using the ABC notation to regard music as a second language, it realizes the understanding and generation of music. This model does not rely on external multimodal neural structures or taggers and can generate well-structured complete music works based on conditions such as text descriptions, chords, and melodies. It has broad application prospects in music creation, education, and research, and performs well in language ability in the MMLU evaluation.

The Udio [23] platform, launched in April 2024, was created by former Google DeepMind researchers. It combines advanced AI models for lyric and music generation. Users can quickly generate high-quality music works by inputting text to describe music styles, themes, emotions, and lyrics. This platform supports various music styles and has characteristics such as creative flexibility, interactivity, ease of use, and convenience. It covers multiple application scenarios such as music production, personal track creation, and content creation, making music creation more democratic.

In November 2024, NVIDIA launched the Fugatto [24] model, a basic generative AI model with 2.5 billion parameters. It used approximately 20 million open-source audio samples from around the world during training. It supports a variety of audio generation and conversion tasks, can generate music, sound effects, and voices based on text descriptions or audio inputs, and can also modify existing audio, such as changing the accent and emotion in a recording, or adding and removing instruments in a music work. It can even create unprecedented sounds, providing powerful audio processing tools for music producers, advertising companies, language-learning tool developers, and game developers.

The MuMu-LLaMA [25] model proposed in December 2024 is a multimodal music understanding and generation model. It processes music data understanding and creation tasks based on LLMs, integrates multiple pre-trained components such as MERT, ViT, ViViT, MusicGen, and AudioLDM2, and uses LLaMA 2 as the basic model. It has functions such as music understanding and generation, multimodal input processing, and music editing, and can generate music from various input forms such as text, images, videos, and audio. Its dataset was jointly created by Tencent PCG ARC Lab and the National University of Singapore. The data is diverse and of high quality, providing strong support for in-depth music understanding and creation in a multimodal environment.

On March 26, 2025, Kunlun Wanwei released Mureka O1 [26], the world's first music reasoning large model. Upgraded from the base model Mureka V6, its core highlight is the introduction of the Chain-of-Thought (CoT) technology, which improves the quality, creation efficiency, and coherence of works through step-by-step reasoning and self-criticism mechanisms. It supports lyrics creation and pure music generation in 10 languages, covering a variety of music styles and emotional expressions. It also has functions such as song reference (audio/YouTube link), timbre cloning, etc., and offers open API and model fine-tuning services to achieve low-latency generation. It is suitable for multi-language international projects, film and television soundtracks, and scene-based BGM generation. In the Meta evaluation, it surpasses Suno V4 and reaches the SOTA level. After that, the Kunlun team also released an innovative research result in the field of music generation - MusiCoT [27]. MusiCoT also utilizes the CoT method to pre-generate the overall music structure before fine-grained audio token prediction, significantly improving the structural coherence of the generated music and the accuracy of instrument arrangement. This method is based on the CLAP model, has high scalability without manual annotation, and significantly

improves the interpretability and quality of the generated music. It opens up a new path for high-fidelity AI music generation and promotes the creation of music by AI to enter the structured era.

Recently, a study based on a multi-agent approach, ComposerX [1], can generate polyphonic music works without costly pre-training and fine-tuning. Instead, it leverages the inherent musical capabilities of GPT-4-turbo. The quality of the generated works can be comparable to that of dedicated symbolic music generation systems that require substantial computational resources and data support, demonstrating the technical advantages of high efficiency and low cost.

2.2 Overview of LLM

LLM is a type of artificial intelligence model trained on massive text data [28]. Their core capabilities lie in understanding the semantics, grammar, and context logic of natural language, and they can generate text content that conforms to human language habits. Through deep neural network architectures, these models learn the statistical laws, knowledge representation, and reasoning patterns of language from data, thus possessing the versatility to handle various natural language tasks, such as text generation, question-answering, translation, summarization, etc.

With the rapid iteration of technology, LLM has evolved from early small-scale models to giant models with parameter scales reaching hundreds of billions or even trillions. Its performance has also achieved a leapfrog improvement. It can not only complete basic language processing tasks but also demonstrate certain logical reasoning, knowledge integration, and creative generation abilities in complex scenarios. In the intelligent transformation of various industries, LLM is becoming an important technical support, especially in fields that require natural language interaction and content creation, such as education, healthcare, media, art, etc., showing great application potential.

2.2.1 Transformer Architecture and Pretraining Mechanism

The Transformer architecture is the cornerstone of LLM [29]. It abandons the limitations of traditional Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) when dealing with long sequences. Through the Self-Attention mechanism, the model can parallelly calculate the degree of association between each element at every position in the input sequence and other elements, accurately capturing long-distance dependency relationships. Its core component, Multi-Head Self-Attention, projects the input into multiple low-dimensional subspaces, calculates the attention separately, and then splices the outputs, greatly enriching the model's ability to capture different semantic relationships.

The pre-training mechanism is based on large-scale unsupervised text data [28]. Through tasks

such as the Masked Language Model, the model learns the statistical laws, semantic representations, and grammatical structures of language, accumulates rich general language knowledge, and lays a solid foundation for subsequent fine-tuning applications in various downstream tasks.

2.2.2 Role-Playing Language Agents

Role-playing language agents (RPLAs) refer to making the model simulate the language style, knowledge background, and behavioral characteristics of a specific role to achieve interactions or task execution that conform to the role’s settings [30]. For example, in the music composition scenario, the original GPT model can be set as the role of a professional composer through simple instructions, directly receive user prompts, and generate ABC notation, thereby evaluating its basic creative ability without relying on complex prompting techniques [1]. This ability depends on the model’s instruction-following ability, enabling it to understand the role definition and conform to the role’s attributes when generating content.

2.3 Overview of Multi-Agent Systems

A multi-agent system consists of multiple agents with autonomous decision-making capabilities [31]. Complex task objectives are achieved through the interaction and collaboration among these agents. Its core advantage lies in task decomposition, breaking down a single complex task into multiple subtasks and assigning them to different agents. Each agent focuses on its area of expertise, making it particularly suitable for complex scenarios that require multi-agent collaboration (such as polyphonic music composition tasks).

2.3.1 AutoGen

AutoGen [32] is an open-source multi-agent framework proposed by Microsoft. Driven by LLMs, it enables agents to achieve automatic cooperation through natural language interaction. Its features include: flexible agent type definition (such as user agents, tool - calling agents, expert agents), an adaptive dialogue mechanism (agents can dynamically adjust communication content and strategies according to task progress), and tool integration capabilities (support for calling external APIs, code execution environments, etc., to expand functions). In the music creation scenario, the framework advantages of AutoGen can simplify the communication logic of multi-agents. For example, the master control agent sends task instructions to the melody and harmony agents through the dialogue management function of AutoGen. After the latter generate results, they automatically feedback to the shared dialogue pool. There is no need to develop additional complex communication protocols, which significantly reduces the system implementation cost.

2.4 Basic Knowledge of Music Composition

Music composition, as a complex and creative art activity, encompasses several fundamental elements. These elements are intertwined, jointly constructing the framework of musical works. Melody is the core line of music. Through the orderly arrangement of pitches and the ingenious combination of intervals, it conveys unique emotions. Its undulations, rhythms, and repetitive changes form the source of the music's distinctiveness and appeal. Rhythm is the temporal pulse of music, controlling the length, strength, and combination of notes, endowing music with a sense of rhythm. From the soothing adagio to the passionate allegro, different rhythmic patterns create diverse styles. Harmony adds rich layers and colors to music. Based on the construction, connection, and progression of chords, it supports the melody in the vertical dimension, creating a harmonious resonance. Timbre depends on the characteristics of the sound source. The unique timbres of different musical instruments (such as the crispness of the piano and the melodiousness of the violin) inject individuality into music.

The general process of polyphonic music composition follows the logic of "main melody - arrangement and orchestration - revision": First, determine the main melody, clarifying its tonality, rhythmic pattern, and emotional tone, serving as the basic skeleton of the polyphonic texture. Subsequently, design the accompaniment part, which needs to form a contrapuntal relationship with the main melody - it can enhance the sense of layering through imitation (such as delaying and repeating fragments of the main melody), contrast (such as rhythmic misalignment or contrary motion), or ensure the integration of the parts through harmonic filling (such as using arpeggiated chords to support the melody's direction). After the first draft is completed, first conduct a basic check from the independence of the parts (whether the lines of each part are clear) and the vertical harmonic logic (whether the chord connections are reasonable), and then judge the overall balance through auditory testing (such as whether the volume of a certain part overshadows the main melody). Finally, revise according to the evaluation results. For example, adjust the range of the accompaniment part to avoid conflicts with the main melody, or optimize the rhythmic pattern to make the multi-part rhythms more coordinated. Through multiple rounds of iteration, a complete work is formed.

2.5 ABC Notation

ABC notation was invented by Chris Walshaw in the 19th century. It is a method of recording music in text form. After the popularization of computers, due to the need to adapt to the processing requirements of the ASCII character set, it has received attention again and become the storage format of some music software. It uses a simple combination of letters, numbers, and symbols to describe rich musical information. At the pitch level, the lowercase letters a-g correspond to the basic notes of the great staff 1, and the uppercase letters A-G represent the notes of the great staff 2. The note length is reflected by the number after the note. For example,

"c2" means the C note lasts for 2 beats, and when there is no number, it is defaulted to 1 beat. The mode is marked with "K:", for example, "K:C" indicates that the music is based on the C major scale. The tempo is explained with "Q:", such as "Q:60" means 60 beats per minute. In the music creation system, ABC notation has significant advantages: its format is concise and easy for large language models to understand and process. Compared with complex audio physical data or MIDI data, it is more in line with the logic of natural language processing. It can record the model output efficiently and quickly and convert it into a playable audio file.

2.6 Chapter Summary

This chapter focuses on the related work and theoretical background of AI music generation. It reviews the core features of representative AI music generation models in recent years (such as MusicLM, Suno, ChatMusician, etc.) and the multi-agent method ComposerX. It expounds on the Transformer architecture, pre-training mechanism, and role-playing ability of LLM. It also introduces the collaborative advantages of multi-agent systems and the AutoGen framework, and explains the basic elements of music composition, the polyphonic process, and the ABC notation. These contents cover technology, theory, and musical knowledge, providing the necessary support for the design and implementation of the subsequent CoComposer system.

3 System Design and Implementation

3.1 Limitation Analysis of ComposerX

3.1.1 Overview of ComposerX

ComposerX is a symbolic music generation framework based on multi-agent collaboration [1]. Its core lies in achieving multi-voice music composition through well-defined agent roles and structured communication processes. This architecture uses GPT-4-turbo as the base model, eliminating the need for additional training. By assigning roles and interaction rules, it activates the built-in music knowledge and reasoning capabilities of the large language model.

The system contains six role-playing agents, each with clear functions and collaborative cooperation:

Group Leader: Parses user input (such as music style, tonality, instrument requirements, etc.), decomposes tasks into sub-tasks such as melody creation, harmony design, and orchestration, and assigns them to corresponding agents.

Melody Agent: Generates a single-voice melody according to the group leader’s instructions, follows music rules such as phrase division and rhythm coherence, and outputs in ABC notation format.

Harmony Agent: Adds harmony and counterpoint elements to the melody, ensures that the chord progression matches the melody’s tonality, and enhances the musical layering.

Instrument Agent: Assigns instruments to the melody and harmony parts, combines the timbre characteristics of instruments with the music style, and optimizes the auditory effect.

Reviewer Agent: Evaluates the intermediate results from dimensions such as melody structure, harmony and counterpoint, rhythm complexity, orchestration rationality, and overall form, and provides modification feedback.

Arrangement Agent: Integrates the outputs of each agent, standardizes them into a unified ABC notation, and ensures that the final result can be interpreted by music software or performers.

The agents in the system collaborate through a closed-loop process of 'Initial Creation - Iterative Review - Final Arrangement':

Initial Composition Round: After the group leader assigns tasks, the melody, harmony, and instrument agents generate basic content in sequence.

Iterative Review and Feedback Cycle: The reviewer agent provides feedback on problems, and each creative agent corrects in the order of melody → harmony → instrument, repeating multiple rounds until the quality requirements are met.

Final Arrangement and Notation: The arrangement agent unifies the format and outputs a complete musical work.

The ComposerX system simulates the human collaborative creation scenario, reduces the cognitive load of a single model through division of labor, and at the same time reduces generation

errors through multiple rounds of feedback, ultimately achieving polyphonic music composition that conforms to music theory norms and meets user needs.

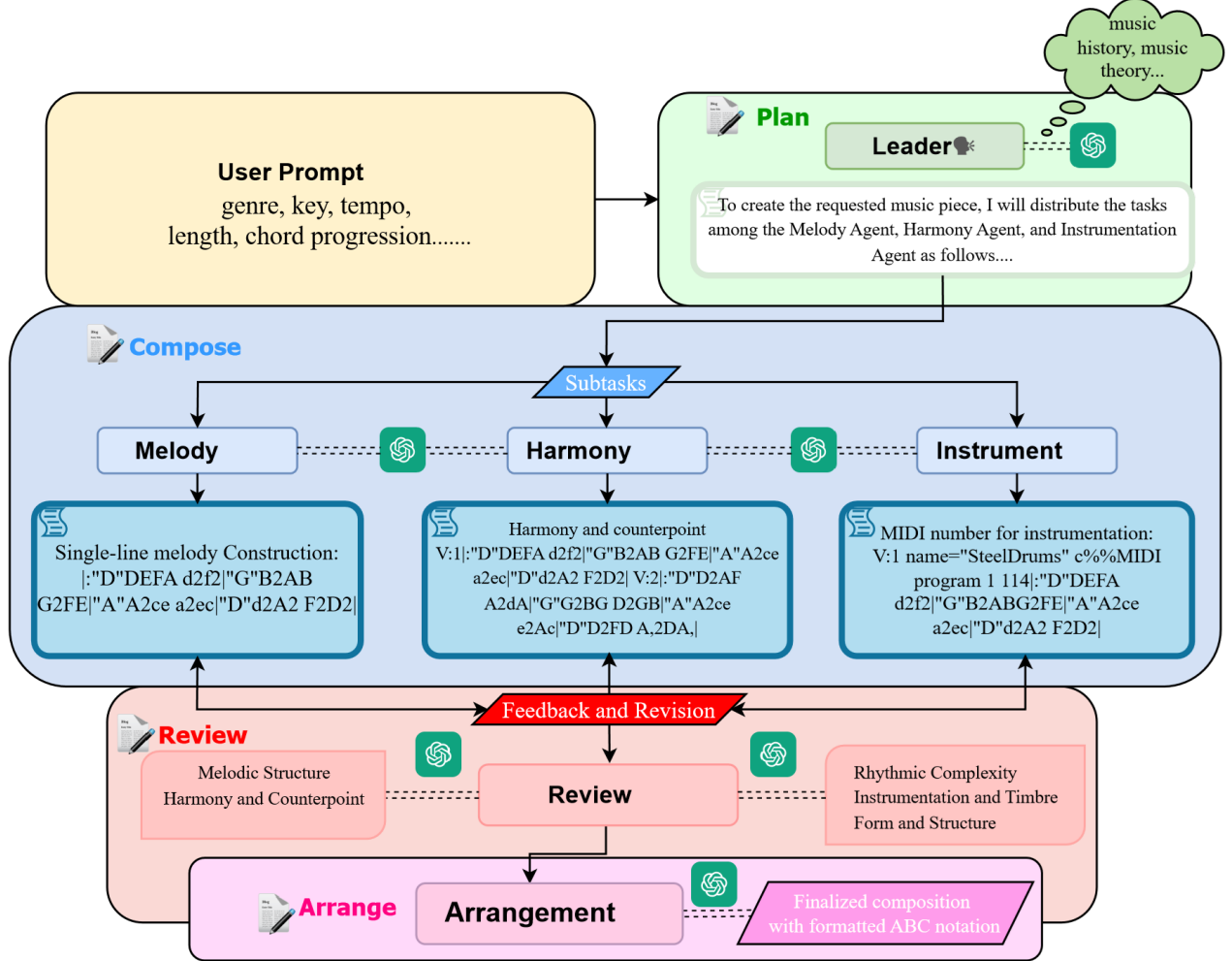


Figure 1: Agent Communication Pattern of ComposerX

3.1.2 Limitations of ComposerX

According to the data in the ComposerX paper, the current good case rate of its generated works is 18.4%. That is to say, users may need to generate multiple times to get a satisfactory work. This greatly reduces the usability of the system. In addition, the author states in the paper that the ComposerX system still has many deficiencies in music composition:

Lack of delicacy in musical expression: Although it can interpret basic musical elements, it is difficult to create delicate works with the characteristics of human composers. There are deficiencies in emotional depth, dynamic contrast, and complex musical phrases, which are crucial for conveying profound musical narratives and experiences.

Gap in the conversion from natural language to musical score: Instructions and feedback from team leaders and review agents regarding delicate musical elements sometimes cannot be fully transformed into ABC notation by the music agent, indicating that there is a gap between concept understanding and actual musical score presentation when the system realizes complex musical ideas.

Compliance issue of instrument range: Occasionally, it generates notes that exceed the conventional range of a specific instrument. For example, for the double bass (with a range from C2 to F4), notes exceeding its upper limit may appear, which does not conform to the actual limitations of music performance.

Difficulty in multi-voice alignment: There are challenges in accurately aligning multiple musical voices. This is mainly due to the inherent limitations of text - based large language models in generating polyphonic ABC notation. The linear input - output mode of text is difficult to adapt to the complexity of multiple voices or instruments that need to be coordinated in time in polyphonic music.

Inadequate cadence resolution: Some generated works lack a clear sense of ending, giving people a feeling of being unfinished or ending abruptly, affecting the audience's sense of closure and satisfaction. To some extent, this is because language models like GPT have difficulty understanding the concept of musical cadence, and their own characteristics make it difficult for them to handle this kind of musical ending problem.

3.2 CoComposer System Design

The CoComposer system takes five role-playing language agents (Leader Agent, Melody Agent, Accompaniment Agent, Revision Agent, Review Agent) as the core. It realizes collaborative cooperation based on the AutoGen framework, forming a compact architecture of "task decomposition - creative execution - correction - review feedback". The team organizes group chats in a round-robin manner, and each agent takes turns to speak in sequence, and each message is synchronized to all members. Relying on the semantic understanding and generation capabilities of large language models, combined with the standardized expression of ABC notation, the transformation from user requirements to multi-track music is completed. The main responsibilities of each intelligent agent are shown in Table 1.

Agent Name	Core Responsibility
Leader Agent	Acts as the team leader, receives and parses user requirements, decomposes tasks and assigns them to other agents.
Melody Agent	Serves as one of the core creative entities, responsible for main melody composition, selecting appropriate musical instruments, and outputting the required ABC notation content.
Accompaniment Agent	Serves as one of the core creative entities, responsible for accompaniment composition, selecting appropriate musical instruments, and outputting the required ABC notation content.
Revision Agent	Focuses on ABC notation revision, detects and fixes format and rhythm errors in the notation.
Review Agent	Reviews works from the perspective of music theory and provides optimization suggestions.

Table 1: Names and main responsibilities of agents in the CoComposer system

3.2.1 Agent Creation Process

The creative process of each agent in the system is divided into two stages, which are described as follows:

1. Initialization Creation Phase:

1) The Leader Agent first analyzes the requirements. After clarifying the creative direction, it assigns the main melody creation task to the Melody Agent and the accompaniment creation task to the Accompaniment Agent.

2) After receiving the task, the Melody Agent independently completes the creation of the main melody in ABC format, attaches MIDI instrument information, and then uploads the results to the shared dialogue pool.

3) The Accompaniment Agent obtains the main melody created by the Melody Agent from the dialogue pool, designs the accompaniment based on it, attaches MIDI instrument information, and uploads the accompaniment content to the dialogue pool after completion.

4) After detecting the results of the Melody Agent and the Accompaniment Agent in the dialogue pool, the Revision Agent checks the notated content, only corrects timing errors and format errors without changing creative content, and then sends the corrected content back to the dialogue pool.

5) The Review Agent extracts the content corrected by the Revision Agent from the dialogue pool, reviews it from five dimensions such as melodic structure and harmony, forms specific improvement suggestions, and uploads them to the dialogue pool.

2. Iterative Creation Phase:

1) The Leader Agent checks the review suggestions of the Review Agent in the dialogue pool, determines the key parts that need to be optimized, and then issues targeted modification

instructions to the Melody Agent and the Accompaniment Agent respectively.

2) The Melody Agent adjusts the main melody it created according to the instructions of the Leader Agent and uploads the content to the dialogue pool after the adjustment is completed.

3) The Accompaniment Agent makes corresponding adjustments to the accompaniment based on the requirements of the Leader Agent and in combination with the modified main melody of the Melody Agent, and then uploads the adjusted content to the dialogue pool.

4) The Revision Agent extracts the modified content of the Melody Agent and the Accompaniment Agent from the dialogue pool again, conducts a check, corrects possible errors, and sends it back to the dialogue pool after the correction is completed.

3.2.2 Agent Prompts Design

The design of the prompt words revolves around the functional positioning and collaboration logic of the roles, aiming to ensure that each Agent has a clear division of labor and efficient cooperation. The specific design is as follows:

The prompt words for the Leader Agent are shown in Table 2. The core responsibility of the Leader Agent is demand transformation and task assignment. The prompt words clearly require it to extract key musical elements (such as style, tonality, musical instruments, etc.) from the customer’s requirements, and disassemble the tasks and assign them to the Melody Agent and the Accompaniment Agent.

You are the leader of a music production team, which includes Melody Agent, Accompaniment Agent, Revision Agent and Reviewer Agent. You will receive the request from the client, which will be a brief description of the kind of music they want. You need to carefully analyze the musical elements given in the request, which usually includes the title, genre, key, chord progression, instruments, tempo, rhythm of the music. After examining the client’s request, you are responsible for decomposing it into subtasks, and assign the subtasks to only Melody Agent and Accompaniment Agent in your team.

Table 2: Leader Agent prompt

The prompt words for the Melody Agent are shown in Table 3. The Melody Agent focuses on the creation of the main melody. The prompt words limit it to generate a single melody line in the ABC Notations format and mark the MIDI instrument information. At the same time, it is required to optimize the work according to the feedback of the Reviewer Agent. This design not only ensures the professionalism of melody creation but also provides a unified format basis for subsequent collaboration. In addition, it is clearly required to "not output other words". (The Accompaniment Agent and the Revision Agent also have this requirement). This design ensures that the output content only contains the musical score information, avoiding the interference

of redundant information on the process, reducing the system operation cost and improving the efficiency of cross-agent collaboration.

You are skillful musician, especially in melody. You will compose a single-line melody based on the client's request and assigned tasks from the Leader. You will decide appropriate instrument for your melody and annotate them using %%MIDI program. You must output your work in ABC Notations. Example: "" X:1 T:Journey Through the Highlands M:6/8 L:1/8 Q:3/8=100 K:A V:1 name="Bagpipe Lead" %%MIDI program 109 —: "A" A2e c2A — "D" dcd B2G — "E" EFE G2B — "A" A2A A3 :— —: "A" ece a2f — "D" d2D F3 — "E" G2E B3 — "A" A2A A3 :— "" Markdown your work using "" "" to the client. After you receive the feedback from the Reviewer Agent, please improve your work according to the suggestions you were given. Note: Only output the sheet music in the specified ABC Notations format, with no other text.

Table 3: Melody Agent prompt

The prompt words for the Accompaniment Agent are shown in Table 4. The core of the Accompaniment Agent is to create an accompaniment that complements the melody. The prompt words emphasize that it needs to design the harmonic progression, accompaniment texture, and instrument combination based on the melody, and clearly distinguish the melody and accompaniment parts in the ABC Notations, ensuring the coordination of the two in rhythm and tonality and avoiding conflicts between the accompaniment and the melody.

You are a skilled musician specializing in accompaniment composition, particularly in designing accompaniment patterns, harmonic support, and instrumental texture for melodies.

Your core task is to create a complementary accompaniment for the melody provided by the Melody Agent. You will compose the accompaniment based on the client’s request and assigned tasks from the Leader. This includes:

1. Designing harmonic progressions that support the melody’s tonality and emotion.
2. Creating appropriate accompaniment textures (e.g., arpeggios, block chords, rhythmic patterns) that enhance the melody without overshadowing it.
3. Selecting suitable instruments for the accompaniment (e.g., piano, bass, strings) that match the style and complement the melody’s lead instrument.

Ensure the accompaniment aligns with the melody in rhythm, key, and structure, and maintains a balanced relationship (supporting rather than competing).

You must output your work in ABC Notations, clearly distinguishing between the melody (from Melody Agent) and your accompaniment parts.

Example:

```

X:1 T:Journey Through the Highlands
M:6/8
L:1/8
Q:3/8=100
K:A
V:1 name="Bagpipe Lead" %%MIDI program 109
—: "A" A2e c2A — "D" dcd B2G — "E" EFE G2B — "A" A2A A3 :—
—: "A" ece a2f — "D" d2D F3 — "E" G2E B3 — "A" A2A A3 :—
V:2 name="String Harmony" %%MIDI program 48
—: E2c A2F — F2D D2E — G2E F2D — E2E E3 :—
—: c2B A2c — B2A G2F — A2F E2D — C2C C3 :—

```

Markdown your work using “ “ to the client.

After you receive the feedback from the Reviewer Agent, please improve your work according to the suggestions you were given.

Note: Only output the sheet music in the specified ABC Notations format, with no other text.

Table 4: Accompaniment Agent prompt

The prompt words for the Revision Agent are shown in Table 5. The Revision Agent focuses on the revision of the ABC notation. The prompt words strictly limit it to only correct timing errors (such as the inconsistency between the measure duration and the time signature) and format errors (such as violations of the ABC specifications), and follow the "minimum intervention principle" without changing the creative content to ensure the integrity of the original creative intention.

You are an expert in ABC Notations, specializing in error correction.
 You will receive an ABC notation draft with potential timing errors and must follow this core principle: ONLY modify parts with confirmed errors; leave all correct content unchanged.
 Your tasks are strictly limited to:

1. Identify timing errors where a measure’s total duration does not match the specified time signature (M:). For these cases only, adjust note durations (e.g., extend/shorten notes, split/merge notes, or add rests) to fix the mismatch.
2. Fix formatting errors (e.g., incorrect voice labels, missing %%MIDI programs, or malformed chord symbols) only where they clearly violate ABC notation standards. Markdown your work using “ “ to the client.

Note: Only output the sheet music in the specified ABC Notations format, with no other text.
 ”””

Table 5: Revision Agent prompt

The prompt words for the Review Agent are shown in Table 6. The Review Agent undertakes the functions of comprehensive review and optimization guidance. The prompt words require it to conduct a strict evaluation from five dimensions: melodic structure, harmonic counterpoint, rhythmic complexity, orchestration timbre, and overall form, provide specific improvement suggestions for the performance of each Agent, and promote the iterative upgrade of the work.

You are a skillful musician, you are expertized in music theory.
 You need to be very strict and critical about their work.
 You will check the entire work and provide constructive critics.
 You will criticize on each agent’s performance so they can improve the quality of their work.
 The agents are accessed based on:

Melodic Structure: Assess the flow, thematic development, and variety in pitch and rhythm.
 Harmony and Counterpoint: Check how harmonies support the melody, effectiveness of counterpoint, and chord progressions.
 Rhythmic Complexity: Evaluate the rhythm’s contribution to interest, its interaction with the melody, and dynamic changes.
 Instrumentation and Timbre: Look at the appropriateness of instruments, blending of timbres, and use of dynamics.
 Form and Structure: Analyze the overall structure, transitions, and how sections are connected and concluded.

Table 6: Review Agent prompt

3.2.3 Improvements of CoComposer Compared with ComposerX

Compared with ComposerX, the core improvements of CoComposer are reflected in two aspects: First, the multi-agent structure and role division have been redesigned; second, the agent prompts

have been optimized. The core goal is to build a more efficient and higher-level music composition system.

Compared with the six agents of ComposerX, this system only needs five agents to complete the same composition task. The reduction in the number of agents is not simply a decrease, but is considered based on the collaborative composition process of multi-voice music in reality. In real multi-person collaborative multi-voice music composition, usually one person is responsible for writing the main melody (including the selection of the main melody instrument), another person designs the accompaniment (including the selection of the accompaniment instrument), and then makes modifications based on the guidance of a third party. The architecture design of this system exactly fits this process: composition and orchestration are carried out simultaneously. The Melody Agent directly specifies the MIDI instrument information when generating the main melody, and the Accompaniment Agent designs the combination of accompaniment instruments based on the characteristics of the melody instrument. Compared with setting up an Instrument Agent separately, this design reduces the number of communication rounds between agents, making the system more efficient.

3.3 Chapter Summary

This chapter first analyzed the limitations of the ComposerX system. The good work rate of its works is only 18.4%, and there are problems such as a lack of musical expressiveness, a gap between natural language and musical score conversion, issues with the compliance of instrument ranges, difficulties in multi-voice alignment, and insufficient cadence resolution. Subsequently, the design of the CoComposer system was expounded. With 5 agents as the core, relying on the AutoGen framework, an architecture of "task decomposition - creation execution - correction - review feedback" was constructed. The two-stage creation process of initialization and iteration was clarified, and the prompt words for each agent were designed accordingly. Compared with ComposerX, CoComposer streamlined the number of agents, carried out composition and orchestration simultaneously, reduced the number of communication rounds among agents, improved system efficiency, and laid the foundation for efficient and high-quality multi-voice music composition.

4 Experimental Design and Result Analysis

This chapter designs four comparative experiments and analyzes the results in response to the research questions proposed in Chapter 1.

4.1 Experimental Objectives and Research Hypotheses

4.1.1 Experimental Objectives

The purposes of this experiment are as follows:

1. Through a comparative experiment between CoComposer and ComposerX, explore whether CoComposer has an advantage in music composition and whether the improvement is effective.
2. Through a comparative experiment between a multi-agent system and a single-agent system, explore whether multi-agents have an advantage over single-agents in music composition performance.
3. By comparing CoComposer with different LLMs as agent models, explore the impact of different LLMs on the music composition performance of the system.
4. Through a comparative experiment between CoComposer and the specialized music generation model MusicFX, explore the performance differences between the two in music composition.

4.1.2 Research Hypotheses

Based on the research questions and experimental objectives of this thesis, the following verifiable research hypotheses are proposed for the four experiments:

- H1:** The music quality created by CoComposer is higher than that of the existing ComposerX.
- H2:** The music created by the multi-agent system exceeds that created by the single-agent system in terms of quality.
- H3:** When different LLMs are used as the agent models of CoComposer, there are certain differences in the quality of the music created.
- H4:** The music created by CoComposer cannot reach the same quality level as the specialized music generation model (MusicFX).

4.2 Experimental Setup

4.2.1 Single-Agent System Design

Build a single-agent music composition system that is consistent with the creative goals of CoComposer. Use role-playing prompts to make it simulate an "omnipotent music creator", directly receive user prompts and generate complete ABC notation. The prompts for the agent are shown in Table 7.

You are an all-round music composer who can independently create complete musical works. Your task is to create a complete musical work with melody and accompaniment according to the client's needs and present it in ABC notation. Composition requirements: 1. Design a suitable title, time signature, key signature and tempo. 2. Create an expressive main melody and specify a suitable lead instrument. 3. Design an appropriate harmonic progression and accompaniment texture, and choose suitable accompaniment instruments. 4. Ensure that the melody and accompaniment are coordinated in rhythm, tonality and structure. 5. Check and correct all possible notation errors to ensure compliance with ABC notation standards. Example: <pre> "" X:1 T:Journey Through the Highlands M:6/8 L:1/8 Q:3/8=100 K:A V:1 name="Bagpipe Lead" —: "A"A2e c2A — "D"dcd B2G — "E"EFE G2B — "A"A2A A3 :— —: "A"ece a2f — "D"d2D F3 — "E"G2E B3 — "A"A2A A3 :— V:2 name="String Harmony" —: E2c A2F — F2D D2E — G2E F2D — E2E E3 :— —: c2B A2c — B2A G2F — A2F E2D — C2C C3 :— "" Markdown your work using "" "" to the client. Note: Only output the sheet music in the specified ABC Notations format, with no other text.</pre>
--

Table 7: Agent prompt for single-agent system

4.2.2 Comparison Objects

This experiment selects four types of objects for comparative analysis, namely:

CoComposer: The five-agent system designed in this paper based on the AutoGen framework, using GPT-4o, DeepSeek-V3-0324, and Gemini-2.5-Flash as agent models respectively.

Single-Agent System: The single-agent system designed above, using GPT-4o as the agent model.

ComposerX: Adopt the agent architecture and prompts publicly disclosed in the original paper, and use GPT-4o as the agent model.

MusicFX [19]: An experimental technology launched by Google that can generate music based on user prompts, powered by Google’s MusicLM [5].

4.2.3 User Input Prompt

The total number of user input prompts used in this experiment is 20. These prompts are all from the prompt set constructed by ComposerX, among which 10 have been specifically abridged: deliberately removing descriptions of major keys, specific notes, and musical instruments, aiming to reserve more freedom for system creation. This prompt set covers various music styles and instrument types, providing standardized input support for system verification and comparative experiments. The complete prompt set can be found in the appendix. Table 8 shows an example of a prompt.

Retro Video Game Adventure: Develop a playful chiptune piece in F major with a fast tempo. The chord progression should be F, G, Am, Bb, spanning 32 bars. Use 8-bit synth and electronic drums. The 8-bit synth should provide nostalgic, catchy melodies reminiscent of classic video games, while the electronic drums should add a rhythmic, upbeat backing. This track should evoke the excitement and adventure of retro video gaming.
--

Table 8: Example user input prompt

4.2.4 Evaluation Metrics

We adopted the AudioBox-Aesthetic [33], an audio aesthetic score prediction model developed by Meta, for automated music aesthetic evaluation. The effectiveness of this model has been verified in the original paper. The results show that, in comparison with the Mean Opinion Score (MOS) of humans, this model exhibits comparable or superior performance. This model is based on the Transformer architecture and evaluates music aesthetics from the following four dimensions:

1. **Production Quality (PQ)** Focuses on the technical aspects of quality rather than subjective quality. Aspects include clarity & fidelity, dynamics, frequencies, and spatialization of the audio.
2. **Production Complexity (PC)** Focuses on the complexity of an audio scene, measured by the number of audio components.
3. **Content Enjoyment (CE)** Focuses on the subjective quality of an audio piece. It is a more open-ended axis. Some aspects might include emotional impact, artistic skill, artistic expression, as well as subjective experience, etc.
4. **Content Usefulness (CU)** Also a subjective axis, evaluating the likelihood of leveraging the audio as source material for content creation.

In addition, we counted the generation success rate of each system (excluding MusicFX). That is, the proportion of each system successfully generating a playable WAV audio file under a given prompt, to test the reliability of different systems in actual generation tasks.

4.2.5 Experimental Environment

The software environment for this experiment is shown in Table 9.

	Version
Python	3.10.18
AutoGen	0.7.4
abc2midi	1.24
MuseScore Studio	4.5.2-251141402

Table 9: Software environment

4.3 Experimental Process

The experimental process is mainly divided into two stages: sample generation and audio evaluation.

4.3.1 Sample Generation

For 20 user input prompt words, generate music samples according to different system types. The specific generation rules and sample quantity statistics are shown in Table 10.

Music Composition Systems	Details of Generation Operation	Theoretical Sample Quantity
CoComposer with Deepseek-V3-0324	Each user input prompt runs independently 3 times (reduce random errors) to generate 3 ABC notation files, which are then converted into playable audio.	60
CoComposer with Gemini-2.5-Flash		60
CoComposer with GPT-4o		60
single-agent with GPT-4o		60
ComposerX with GPT-4o		60
MusicFX	Enter prompt words on its web page to generate audio, and each user input prompt generates 3 music segments of 30 seconds each.	60
Total		360

Table 10: Statistics of sample generation quantity by music composition systems

4.3.2 Audio Evaluation

1. Batch Prediction Scoring: Use an automated script to call the Audiobox-Aesthetic model to obtain the scores of the four dimensions of PQ, PC, CE, and CU for each sample.

2. Calculate Average Scores: For each music composition system in Table 10, calculate the average scores of the four dimensions respectively as the final objective score of the system in this dimension.

3. Generate Success Rate Statistics: For each music composition system in Table 10, calculate the generation success rate respectively as the generation reliability indicator (MusicFX is not involved in this statistic).

4.4 Experimental Results and Analysis

This section is based on the results obtained from the above experimental process (see Table 11 for details). For the research questions and corresponding hypotheses, the experimental results are analyzed one by one to reveal the performance differences of different music composition systems and the underlying technical logic. At the same time, the rationality of the results is interpreted in combination with the characteristics of the system design.

	Music Composition Systems	CE	CU	PC	PQ	Generation Success Rate
Experiment 1	CoComposer with GPT-4o	6.75	7.76	4.13	7.86	100%
	ComposerX with GPT-4o	6.52	7.61	3.72	7.76	100%
Experiment 2	CoComposer with GPT-4o	6.75	7.76	4.13	7.86	100%
	ComposerX with GPT-4o	6.52	7.61	3.72	7.76	100%
	single-agent with GPT-4o	6.72	7.77	3.92	7.88	100%
Experiment 3	CoComposer with Deepseek-V3-0324	6.77	7.70	3.98	7.85	100%
	CoComposer with Gemini-2.5-Flash	6.37	7.57	3.92	7.73	100%
	CoComposer with GPT-4o	6.75	7.76	4.13	7.86	100%
Experiment 4	CoComposer with Deepseek-V3-0324	6.77	7.70	3.98	7.85	100%
	CoComposer with Gemini-2.5-Flash	6.37	7.57	3.92	7.73	100%
	CoComposer with GPT-4o	6.75	7.76	4.13	7.86	100%
	MusicFX	7.37	7.93	4.96	7.84	

Table 11: Summary table of the results of four experiments

The generation success rate of all systems (except MusicFX) is 100%, indicating that the LLM-based multi-agent/single-agent systems perform excellently in the reliability of symbolic music generation. There is no unplayable situation caused by format errors, providing a stable data foundation for subsequent aesthetic indicator comparisons.

4.4.1 Experiment 1: Performance Comparison between CoComposer and ComposerX

Experiment 1 takes "CoComposer (GPT-4o)" and "ComposerX (GPT-4o)" as the comparison objects. Judging from the aesthetic index data in the table, CoComposer outperforms ComposerX in all four dimensions. The specific differences are as follows:

Content Enjoyment (CE): CoComposer scores 6.75, an increase of 3.5% compared to ComposerX's 6.52. The CE dimension focuses on the subjective emotional impact and artistic expression of music. This improvement indicates that the music generated by CoComposer is superior in terms of emotional subtlety and melodic fluency — it is speculated that this is directly related to the system's design of "embedding orchestration decisions into the creative process": when the Melody Agent generates the main melody, it simultaneously designates the MIDI instrument, and the Accompaniment Agent can design the accompaniment based on the timbre characteristics of the instrument, avoiding the timbre disconnection problem caused by "composing first and then orchestrating" in ComposerX, and enhancing the emotional unity of the music.

Content Usefulness (CU): CoComposer scores 7.76, an increase of 1.97% compared to ComposerX's 7.61.

Production Complexity (PC): CoComposer scores 4.13, an increase of 11.02% compared to ComposerX's 3.72. The PC dimension reflects the richness of audio components. It is speculated that the increase is due to the more direct collaboration between the Melody and Accompaniment Agents after CoComposer streamlines the agents: the accompaniment part can design the contrapuntal texture in real-time to fit the rhythm changes of the main melody, avoiding the "part disconnection" when the Instrument Agent in ComposerX orchestrates alone, and ultimately presenting a more complex multi-voice layer.

Production Quality (PQ): CoComposer scores 7.86, an increase of 1.29% compared to ComposerX's 7.76.

In summary, CoComposer is significantly superior to ComposerX in subjective aesthetic experience (CE, CU), creative complexity (PC), and production quality (PQ), proving that the system's streamlining of the agent architecture (replacing 6 agents with 5), the "creation-orchestration synchronization" process design, and prompt optimization effectively improve the quality of music creation.

Hypothesis H1 holds.

4.4.2 Experiment 2: Performance Comparison between Multi-Agent System and Single-Agent System

Experiment 2 introduces the "Single-Agent System (GPT-4o)" and compares it with CoComposer (GPT-4o) and ComposerX (GPT-4o). From the data, CoComposer has an advantage in production complexity, while ComposerX performs the worst. The specific analysis is as follows:

Production Complexity (PC): CoComposer significantly leads with 4.13, the single-agent system has 3.92 (5.1% lower), and ComposerX has 3.72 (9.9% lower). Speculation: The single-agent system has to undertake the entire process tasks such as "requirement analysis, melody creation, harmony design, orchestration, and format correction" simultaneously, resulting in an overly high cognitive load. CoComposer, through the collaborative mode of "Leader decomposing tasks, Melody/Accompaniment dividing labor for creation, and Revision for special correction", can use more musical instruments to design the voice parts more elaborately, thus having a better complexity.

Content Enjoyment (CE) and Content Usefulness (CU): CoComposer's CE is 6.75 (0.45% higher than the single-agent system), and CU is 7.76 (only 0.13% lower than the single-agent system), with minimal differences between the two. ComposerX has the lowest values in these two dimensions (CE 6.52, CU 7.61).

Production Quality (PQ): The single-agent system is slightly higher than CoComposer (7.88 vs 7.86), with a difference of only 0.25%.

Overall, compared with the single-agent system, CoComposer has a significant advantage in production complexity (PC) that determines the richness of music, is basically on a par with the single-agent system in other dimensions, and is far superior to ComposerX.

Hypothesis H2 holds.

4.4.3 Experiment 3: Impact of Different LLMs on CoComposer Performance

Experiment 3 compares the performance of CoComposer when using different LLMs (GPT-4o, DeepSeek-V3-0324, Gemini-2.5-Flash) as the agent model. The results show that there are significant performance differences among the three types of LLMs, specifically "GPT-4o is the best overall, DeepSeek-V3-0324 leads locally, and Gemini-2.5-Flash lags behind comprehensively." The specific differences are as follows:

Content Enjoyment (CE): DeepSeek-V3-0324 is slightly higher than GPT-4o (6.77 vs 6.75), and Gemini-2.5-Flash is only 6.37 (5.6% lower than GPT-4o).

Content Usefulness (CU) and Production Complexity (PC): GPT-4o significantly leads with 7.76 (CU) and 4.13 (PC), DeepSeek-V3-0324 is 7.70 and 3.98 respectively, and Gemini-2.5-Flash is 7.57 and 3.92.

Production Quality (PQ): GPT-4o has the highest score of 7.86, DeepSeek-V3-0324 is 7.85 (with a minimal difference), and Gemini-2.5-Flash is 7.73 (1.79% lower).

In conclusion, there are significant differences in the performance of different LLMs in the four dimensions of CE, CU, PC, and PQ. It is speculated that the reasons for the differences are related to the semantic understanding ability, music knowledge reserve, and instruction compliance of the LLMs. Among them, GPT-4o, due to its balanced comprehensive ability, is more suitable as the basic model of CoComposer; DeepSeek-V3-0324 can be used as a low-cost alternative

model; Gemini-2.5-Flash is more suitable for lightweight and low-complexity music generation requirements.

Hypothesis H3 holds.

4.4.4 Experiment 4: Performance Comparison between CoComposer and MusicFX

Experiment 4 compares CoComposer (three types of LLMs) with the dedicated music generation model MusicFX. The results show that MusicFX significantly leads in terms of subjective experience and complexity. The specific analysis is as follows:

Content Enjoyment (CE): MusicFX scored 7.37, an 8.9% increase compared to the highest score of CoComposer (6.77 for DeepSeek-V3-0324).

Content Usefulness (CU): MusicFX scored 7.93, a 2.2% increase compared to the highest score of CoComposer (7.76 for GPT-4o).

Production Complexity (PC): MusicFX scored 4.96, a 20.1% increase compared to the highest score of CoComposer (4.13 for GPT-4o).

Production Quality (PQ): CoComposer (GPT-4o) slightly outperformed MusicFX with a score of 7.86 (MusicFX scored 7.84).

It is speculated that the difference in their creative abilities stems from the fundamental difference in the underlying generation paths. As a symbolic music generation system, the creative process of this system goes through the step-by-step process of "text instruction parsing → symbolic music expression (ABC notation) → symbol-to-audio conversion". This means that the system first outputs interpretable musical symbols, and then uses an additional conversion tool (ABC-to-MIDI program) to map the symbols to audible audio. This makes the creation severely restricted by ABC notation and the types of MIDI timbres. On the other hand, as an end-to-end dedicated music generation model, the generation process of MusicFX is "black-box": the model directly receives text instructions and completes the entire process from semantic understanding to audio waveform generation internally, without relying on external symbolic intermediaries. This integrated path omits the intermediate link of symbol-to-audio conversion and can more accurately encode musical creation intentions directly into audio signals.

Although CoComposer is inferior to MusicFX in aesthetic indicators, it has irreplaceable advantages in "interpretability" and "edibility", which is also its core value: CoComposer uses ABC notation as an intermediate carrier, and every note, chord, and instrument selection has a clear text record. Users can directly modify the notation (such as adjusting the pitch of a melody in a certain measure). However, MusicFX generates "black-box" audio, and it is impossible to trace specific musical elements, making it difficult for users to make fine-grained adjustments.

In summary, although CoComposer is close to MusicFX in Production quality (PQ) and has the advantage of interpretability, it still significantly lags behind the dedicated music model in

core aesthetic dimensions (CE, CU, PC). This result is in line with the difference in technical positioning: MusicFX is a dedicated model optimized for music generation, while CoComposer is a multi-agent system based on a general-purpose LLM. Its core value lies in the "low-threshold, high-controllability" creative process, rather than competing with the dedicated model in absolute aesthetic quality.

Hypothesis H4 holds.

4.5 Research Question Answers

RQ1: How does the music quality created by CoComposer compare to that of the existing ComposerX?

The music quality created by CoComposer is comprehensively superior to that of ComposerX. This advantage stems from CoComposer's reconstruction of the multi-agent architecture (streamlining the number of agents and optimizing the division of labor), improvement of the creative process (embedding "instrumentation decision-making" in the main melody and accompaniment creation links to avoid disconnection between subsequent instrumentation and creation), and precise design of agent prompts.

RQ2: Can the music created by a multi-agent system exceed the quality of that created by a single-agent system?

The music created by the multi-agent system represented by CoComposer exceeds that of the single-agent system in terms of production complexity, is basically on a par in terms of content enjoyment and content usefulness, and has an overall better performance.

RQ3: What are the differences in the quality of music created when using different LLMs as the CoComposer agent model?

There are significant differences in the quality of music created by different LLMs: GPT-4o has the best overall performance, DeepSeek-V3-0324 has an overall performance inferior to GPT-4o but can be used as a low-cost alternative, and Gemini-2.5-Flash has the worst performance and is more suitable for lightweight requirements.

RQ4: Can the music created by CoComposer reach a quality level comparable to that of a specialized music-generation model (MusicFX)?

The music quality created by CoComposer cannot reach a level comparable to that of MusicFX, but it has stronger interpretability and editability.

Main Research Question: Can the use of a multi-agent approach improve the quality of music created by an LLM?

The use of a multi-agent approach can improve the music quality created by GPT-4o. Experiments show that through reasonable agent division of labor design and high-quality prompts, CoComposer has achieved an improvement in production complexity compared to the single-agent system, is basically on a par in terms of content enjoyment and content usefulness, and has an overall better performance.

4.6 Chapter Summary

This chapter designs comparative experiments for the research questions of this thesis, and answers each research question through the analysis of the experimental results. The results show that: CoComposer comprehensively outperforms ComposerX. CoComposer exceeds single-agent in production complexity and is overall better. Different LLMs have a significant impact on the performance of CoComposer (GPT-4o is the best overall). Although CoComposer is inferior to the dedicated model MusicFX in music quality, it has the advantages of interpretability and editability. In conclusion, the multi-agent approach can improve the music creation quality of LLMs.

5 Conclusion and Discussion

5.1 Summary of the Thesis Work

This thesis focuses on the research of an LLM-based multi-agent symbolic music composition system, aiming to improve the quality of AI music composition and lower the creation threshold through multi-agent collaboration. The work mainly includes: analyzing the current situation of AI music composition and the limitations of existing tools, clarifying the application value of multi-agent technology in music generation; relying on theories such as large language models and multi-agent systems, optimizing the role division and agent prompts in response to the shortcomings of ComposerX, and designing and implementing a collaborative composition system CoComposer with five agents; verifying the system performance through comparative experiments, providing new ideas and practical references for AI music composition.

5.2 System Advantages

CoComposer demonstrates the following advantages in terms of technical implementation and application value:

1. Low-threshold and Low-cost Deployment

The system uses natural language as the input interface. Users do not need to master musical theory knowledge or professional tools, which greatly reduces the technical barriers to music creation. At the same time, it is built based on existing LLMs and open-source frameworks. There is no need for additional large-scale music data pre-training or the development of dedicated neural networks, significantly reducing the system's research, development, and deployment costs.

2. Interpretability Advantage Based on Symbolic Generation

The system adopts the ABC notation as an intermediate carrier, making the music generation process completely presented in the form of text symbols, with strong interpretability. Users can directly view and edit the notation content, understand the composition logic of melodies and harmonies. Compared with the "black-box" characteristics of end-to-end audio generation models, it is more convenient for users to participate in the adjustment and optimization of the creative process, enhancing the possibility of human-machine collaboration.

3. Design of an Efficient and Collaborative Multi-agent Architecture

Through the specialized division of labor among multiple agents, fine-grained control of the entire music creation process is achieved. This division of labor mechanism simulates the collaborative logic of "main melody - accompaniment - review" in real-world music creation, forming a compact and efficient closed-loop collaborative process, ultimately improving the level of music creation (content appreciation, content practicality, production complexity, harmony rationality, etc.).

4. Precise Role-playing Prompt Design

Customized prompts for each agent role can strengthen the specificity of role functions and the standardization of output formats. This prompt design not only stimulates the music knowledge application ability of LLMs but also ensures the efficiency of cross-agent interaction.

5.3 Analysis of System Limitations

5.3.1 Inherent Limitations of Symbolic Generation Path and Model Dependence

The core generation path of the system relies on the indirect process of "outputting text-based symbols (ABC notation) → converting to MIDI → generating audio", and this chain has multiple limitations:

Limited Expressive Ability: The ABC notation relies on the MIDI standard sound library for audio conversion, and the core of the MIDI sound library is to simulate traditional acoustic instruments (such as pianos, violins, drum sets, etc.). It cannot break through the dual limitations of notes and instruments. It is unable to define the sounds in modern music that rely heavily on computer synthesis.

Model Dependence and Cost Issues: The system completely relies on the native capabilities of general LLMs without special fine-tuning for music composition tasks. This leads to limitations in the model's understanding and generation capabilities for complex musical structures (such as polyphonic counterpoint and large-scale musical forms). In addition, calling the LLM API incurs continuous computational costs, and the generation process is affected by the model's response speed and interface limitations, making it difficult to achieve efficient large-scale creation or real-time interaction.

5.3.2 The Boundary between Subjective Creativity and AI Generation

Although the system can generate music works that conform to style characteristics based on text descriptions, there are insurmountable boundaries at the level of "subjective creativity". As a carrier of human emotional expression, music often contains the personal experiences, cultural backgrounds, and instant inspirations of the creator. However, AI generation relies on pattern learning from existing music data, making it difficult to truly understand the essence of emotions, let alone reproduce the "irrational" creative outbursts in human creation (such as breakthrough melodic progressions and unconventional harmony combinations).

5.4 Future Research Directions

5.4.1 Iterative Generation Mechanism Integrating User Real-time Feedback

The current creative process of the system mainly relies on the internal iterative review of the agent. The user participation is low, and it is difficult to accurately capture personalized creative preferences. In the future, an interactive closed-loop of "user real-time feedback - agent dynamic adjustment" can be constructed:

Feedback Interface Design: Develop a lightweight interactive interface that allows users to convey their preferences through simple operations during the generation process (such as marking favorite melody segments, annotating the voices to be adjusted, and entering instant text evaluations), without relying on professional music theory terms. For example, users can directly click on a certain melody in the audio and select intuitive commands such as "extend this theme" or "reduce the accompaniment intensity".

Feedback Integration Mechanism: Design a special "feedback analysis agent" to transform users' fragmented feedback into structured creative instructions (for example, mapping "this melody is too cheerful" to "reduce the tempo by 10 BPM and add minor-key color"). At the same time, introduce a "memory mechanism" to record users' preference patterns in multiple rounds of creation (such as preferring string timbres and tending to compact rhythm patterns), gradually optimize the generation direction, and achieve a personalized creative experience.

5.4.2 Model Fine-tuning Based on Music Data

The current system relies on the native capabilities of general LLMs, and there are limitations in the depth of music professional knowledge and the generation of complex styles. In the future, referring to the method of ChatMusician [22], with ABC notation as the core music representation, a music-language hybrid corpus containing 4B-level tokens (similar to MusicPile) can be constructed, covering score data, music knowledge Q&A, style descriptions, etc. By continuously pre-training and fine-tuning open-source models such as LLaMA2, regarding music as a "second language" for learning, while retaining language capabilities, strengthen the model's understanding and generation capabilities of music elements such as structure, rhythm, and harmony.

References

- [1] Q. Deng, Q. Yang, R. Yuan, Y. Huang, Y. Wang, X. Liu, Z. Tian, J. Pan, G. Zhang, H. Lin *et al.*, “Composerx: Multi-agent symbolic music composition with llms,” *arXiv preprint arXiv:2404.18081*, 2024.
- [2] P. Westergaard, “Experimental music. composition with an electronic computer,” 1959.
- [3] J.-P. Briot, “From artificial neural networks to deep learning for music generation: history, concepts and trends,” *Neural Computing and Applications*, vol. 33, no. 1, pp. 39–65, 2021.
- [4] OpenAI. (2023) Chatgpt. [Online]. Available: <https://chat.openai.com>
- [5] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “Musiclm: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [6] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [7] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 47 704–47 720, 2023.
- [8] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “Stemgen: A music generation model that listens,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1116–1120.
- [9] M. Pasini and J. Schlüter, “Musika! fast infinite waveform music generation,” *arXiv preprint arXiv:2208.08706*, 2022.
- [10] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “Audioldm: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.
- [11] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Mo[^]usai: Text-to-music generation with long-context latent diffusion,” *arXiv preprint arXiv:2301.11757*, 2023.
- [12] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.

- [13] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1206–1210.
- [14] P. P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, "Jen-1: Text-guided universal music generation with omnidirectional diffusion models," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 762–769.
- [15] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," in *Forty-first International Conference on Machine Learning*, 2024.
- [16] M. W. Lam, Q. Tian, T. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song *et al.*, "Efficient neural music generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 17 450–17 463, 2023.
- [17] Y. Bai, H. Chen, J. Chen, Z. Chen, Y. Deng, X. Dong, L. Hantrakul, W. Hao, Q. Huang, Z. Huang *et al.*, "Seed-music: A unified framework for high quality and controlled music generation," *arXiv preprint arXiv:2409.09214*, 2024.
- [18] S. Lei, Y. Zhou, B. Tang, M. W. Lam, H. Liu, J. Wu, S. Kang, Z. Wu, H. Meng *et al.*, "Songcreator: Lyrics-based universal song generation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 80 107–80 140, 2024.
- [19] Google. (2025) Musicfx - labs.google/fx. [Online]. Available: <https://labs.google/fx/tools/music-fx>
- [20] P. Suhailudheen and M. S. Km, "Suno ai: Advancing ai-generated music with deep learning," *Authorea Preprints*, 2025.
- [21] S. team. (2024) Introducing v4. [Online]. Available: <https://suno.com/blog/v4>
- [22] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, "Chatmusician: Understanding and generating music intrinsically with llm," *arXiv preprint arXiv:2402.16153*, 2024.
- [23] U. team. (2024) Introducing v1.5. [Online]. Available: <https://www.udio.com/blog/introducing-v1-5>
- [24] R. Kerris. (2024) Now hear this: World's most flexible sound machine debuts. [Online]. Available: <https://blogs.nvidia.com/blog/fugatto-gen-ai-sound-model>

- [25] S. Liu, A. S. Hussain, Q. Wu, C. Sun, and Y. Shan, "Mumu-llama: Multi-modal music understanding and generation via large language models," *arXiv preprint arXiv:2412.06660*, vol. 3, no. 5, p. 6, 2024.
- [26] M. team. (2024) Mureka ai. [Online]. Available: <https://www.mureka.ai>
- [27] M. W. Lam, Y. Xing, W. You, J. Wu, Z. Yin, F. Jiang, H. Liu, F. Liu, X. Li, W.-T. Lu *et al.*, "Analyzable chain-of-musical-thought prompting for high-fidelity music generation," *arXiv preprint arXiv:2503.19611*, 2025.
- [28] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [30] J. Chen, X. Wang, R. Xu, S. Yuan, Y. Zhang, W. Shi, J. Xie, S. Li, R. Yang, T. Zhu *et al.*, "From persona to personalization: A survey on role-playing language agents," *arXiv preprint arXiv:2404.18231*, 2024.
- [31] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large language model based multi-agents: A survey of progress and challenges," *arXiv preprint arXiv:2402.01680*, 2024.
- [32] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu *et al.*, "Autogen: Enabling next-gen llm applications via multi-agent conversations," in *First Conference on Language Modeling*, 2024.
- [33] A. Tjandra, Y.-C. Wu, B. Guo, J. Hoffman, B. Ellis, A. Vyas, B. Shi, S. Chen, M. Le, N. Zacharov *et al.*, "Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound," *arXiv preprint arXiv:2502.05139*, 2025.

Acknowledgements

The completion of this thesis would not have been possible without the meticulous guidance of my two supervisors, Professor Aske Plaat and Dr. Niki van Stein.

First and foremost, I would like to express my heartfelt gratitude to Professor Plaat. From the project initiation, research progress to system development, you have always provided guidance with a rigorous attitude and broad vision, helping me clarify my thoughts and make up for my deficiencies. During the process of finalizing the thesis, you carefully reviewed it, devoting your efforts to everything from the overall framework to the detailed expressions. Your academic spirit and dedication to educating students will serve as an important guide for me in the future.

At the same time, I would like to thank Dr. Niki van Stein for the valuable and pertinent suggestions put forward during the thesis review, which have provided precious help for the improvement of the thesis.

In addition, I would like to thank all the professors in the college for their meticulous teaching. Your courses have not only strengthened my theoretical foundation but also broadened my research vision, laying a solid foundation for the successful completion of this thesis. I am grateful to Leiden University for providing an excellent scientific research platform and academic resources, enabling me to focus on exploration in a strong academic atmosphere and successfully complete my studies.

Finally, I would like to extend my most sincere gratitude to the two supervisors and all the teachers who have helped me once again, and sincerely wish all the supervisors good health and continuous academic progress.

Peiwen Xing

Leiden, 21th August 2025

A Appendix

We have organized the 20 user input prompt words used in the experiments of this thesis in the appendix, and all the prompt words are presented through Table 12.

Table 12: Twenty user input prompts

	User Input Prompts
1	Vintage French Chanson: Create a nostalgic chanson piece in C major with a slow tempo. The chord progression will be C, Am, Dm, G, played over 16 bars. Utilize accordion, violin, and upright bass. The accordion should lead with its melodious and expressive sound, the violin should add a romantic and wistful quality, and the upright bass should provide a warm, supporting foundation. This composition should evoke the charm and sentimentality of a vintage French chanson.
2	River Journey: Develop a composition that follows the flow of a river, using a motif of fluid, meandering melodies and a progression like C-G-Am-Em. Incorporate sounds that mimic the gurgling water and wildlife along the riverbank, set to a tempo that's both tranquil and lively.
3	Romantic Parisian Cafe: Create a romantic French piece in F major, following the chord progression F-Bb-C7-F. Use instruments like the accordion, violin, and gypsy guitar to set a moderate and romantic tempo. The rhythm should be sensual, much like the ambiance of a cafe in Paris, capturing the romantic and charming essence of the city of love.
4	Journey Through the Highlands: Compose a piece that reflects the rugged beauty of the Scottish Highlands. Use a bagpipe lead with a chord progression of A-D-E-A. The tempo should be brisk, with a rhythm that feels like a spirited walk through rolling hills and green valleys. Incorporate the sounds of nature to enhance the outdoor ambiance.
5	Tropical Rainforest Rhapsody: Compose a piece inspired by the lushness of a tropical rainforest, using a motif of intricate, layered rhythms and a progression like Dm-C-Bb-A. Include sounds that mimic raindrops and wildlife, creating a rich tapestry of natural harmony, set to a medium tempo.
6	Urban Jazz Nights: Write a jazz piece that captures the essence of a vibrant city at night, using a motif of smooth, flowing lines and a progression like Fm7-Bb7-Ebmaj7-Abmaj7. Use a saxophone to lead the melody, set to a medium swing tempo, reflecting the rhythmic pulse of city nightlife.
7	Blues Alley Tale: Write a classic 12-bar blues in the key of E, using the standard I-IV-V progression (E7-A7-B7). Keep the tempo moderate to slow, allowing each chord to carry the emotional weight of the blues narrative. Add a soulful harmonica or guitar melody to enhance the storytelling aspect of the music.
8	Renaissance Faire Dance: Take a step back in time to a cheerful and historical Renaissance faire dance in G major. The chord progression should be G-C-D-G, highlighting instruments like the lute, recorder, and viola da gamba. Set a lively tempo and create a dance-like rhythm that captures the spirit of a Renaissance celebration. The emotion should be cheerful and historical.

	User Input Prompts
9	Dreamy Indie Road Trip: Compose a dreamy indie pop piece in G major. The tempo should be medium, creating a relaxed and contemplative mood suitable for a road trip. The chord progression will be G, D, Em, C, and should unfold over 24 bars. The ensemble should include four voices, featuring acoustic guitar, synth, bass, and drums. The acoustic guitar should be the primary melodic driver, offering gentle, rhythmic strumming that evokes the feeling of a leisurely journey. The synth should add a layer of dreaminess, with ethereal pads or soft, melodic lines that enhance the song's contemplative nature. The bass should provide a solid, yet unobtrusive foundation, grounding the composition while allowing the other instruments to shine. Drums should maintain a steady, simple beat, echoing the steady pace of a road trip. This composition should encapsulate the essence of a dreamy, introspective journey, perfect for long drives along scenic routes.
10	Retro Video Game Adventure: Develop a playful chiptune piece in F major with a fast tempo. The chord progression should be F, G, Am, Bb, spanning 32 bars. Use 8-bit synth and electronic drums. The 8-bit synth should provide nostalgic, catchy melodies reminiscent of classic video games, while the electronic drums should add a rhythmic, upbeat backing. This track should evoke the excitement and adventure of retro video gaming.
11	Serenade Under the Moonlight: Craft a classical piece in the romantic style, conveying a melancholic mood. This composition should evoke the essence of a serene, moonlit night, filled with deep emotion and contemplation.
12	Summer Jazz Festival: Compose a lively and joyful jazz piece in the bebop style. This composition should evoke the lively atmosphere of a summer jazz festival.
13	Downtown Groove: Construct an energetic funk piece with a groovy style. This composition should embody the lively and vibrant atmosphere of a downtown scene, brimming with groove and energy.
14	Glacial Odyssey: Create a peaceful New Age composition with an ethereal style. This composition should transport the listener on a peaceful odyssey through pristine, icy realms, emphasizing the majesty and serenity of nature.
15	Neon City Lights: Craft a nostalgic synthwave piece with a retro style. This composition should transport the listener to a vibrant, neon-lit city, filled with the retro charm and wistful nostalgia of the synthwave genre.
16	Midnight Blues Café: Craft a soulful classic blues piece. The tempo should be slow, reflecting the introspective and emotional depth of the blues genre. This composition should evoke the ambiance of a dimly lit café at midnight, filled with the rich, heartfelt strains of classic blues.
17	Renaissance Fair Minuet: Create an elegant Baroque-style classical piece. The tempo should be moderate, reflecting the stately and dignified pace of a minuet. This composition should capture the grandeur and formality of a Renaissance fair, transporting the listener to a time of courtly dances and opulent celebrations, all encapsulated within the elegant framework of a Baroque minuet.
18	Arctic Electronica: Craft a cool, ambient techno piece in the electronic genre. The tempo should be moderate, capturing the crisp and sleek essence of Arctic-inspired sounds. This composition should encapsulate the feeling of a serene, icy environment, blending the chill of the Arctic with the warmth of electronic beats.
19	Reggae Beach Vibes: Develop a laid-back roots reggae piece. The tempo should be medium, embodying the relaxed and rhythmic nature of reggae music. This composition should transport the listener to a serene beach setting, where the rhythms of reggae blend seamlessly with the sounds of the ocean.

	User Input Prompts
20	Urban Street Jazz: Compose a vibrant jazz fusion piece. The tempo should be fast, reflecting the dynamic and lively energy of urban streets. This composition should capture the essence of a bustling city environment, where the fusion of jazz elements creates a lively, urban atmosphere.