A Comprehensive Investigation of Advances in Music Understanding and Generation Technologies Based on Large Language Models

Jiayi Zhang

St.George's School, Vancouver, Canada
alex.zhang27@stgeorges.bc.ca

Abstract. The intersection of artificial intelligence and music has developed rapidly in recent years, driven by advances in deep learning and the increasing availability of multimodal datasets. This review surveys recent progress in music understanding and generation with Artificial Intelligence (AI) through Large Language Models (LLMs) along three lines: agent/controller systems (e.g., AudioGPT, MusicAgent, CoComposer), multimodal fusion/decoders (MuMu-LLaMA, DeepResonance), and symbolic score models (ChatMusician, MuseCoco). The paper summarizes what each does well, such as tool-orchestrated workflows and cross-modal alignment for text-/image-/video-to-music. Next, the paper introduces the common, fixable challenges occurring in each category, such as limited long-form coherence, uneven controllability, and dataset bias. Some key challenges include orchestration reliability and dependence on pretrained decoders. The paper then proposes some short term remedies such as including multi-agent planning, longer-context modelling, and broader, well-labeled pretraining data, before predicting that the field of music based Large Language Models is moving toward hybrid systems that will integrate themselves within real workflows in the near future. This mapping creates a concise summary of advances in musical understanding of Large Language Models.

Keywords: Large-Language-Model, Artificial Intelligence, Agent-based, Multimodal, Symbolic

1. Introduction

Recent advancements in the field of Large Language Models (LLMS) are reshaping computational approaches to music, from analysis to generation. LLMS are Artificial Intelligence (AI) models that absorb and train on vast amounts of data in order to understand and process human language. As music sequences are amenable to language-modelling techniques, the same methods applied for text are also applicable to musical data. Music understanding in this domain refers to computational tasks that describe or interpret musical content, such as categorizing genres and moods, captioning short clips, and identifying instruments. Meanwhile, Music generation denotes systems that can create musical material in the form of scores or audio.

There have been many successful recent attempts at adapting LLMs to audio and music. A 2024 study carried out by Huang et al. sought to correct LLM processing issues with "managing complex audio inputs and conversational speech" [1] by introducing AudioGPT, a multimodal system that merges a large language model with speech input/output and a toolbox of pre-trained audio models to analyze and generate sound. The model was able to handle several different tasks involving speech and music understanding, utilizing Automatic Speech Recognition (ASR) and Text-to-speech (TTS) to interact with users. A similar study done in late 2024 specifically sought to delve deeper into understanding multi-model music, creating "a multi-modal music understanding and generation model named MuMu-LLaMA" [2]. Taking from an annotated, 167.69 hour long multimodal dataset, the model was able to handle music captioning, text-to-music generation, prompt-based music editing (e.g., tempo/pitch/add-replace operations), and image/video-to-music synthesis within one model family. Several other developments in this field include Chatmusician, which can compose pieces and answer music theory questions, and MusicAgent, which uses a language model to analyze a request before picking external tools to return a final result demo.

Overall, the recent work shows three complementary directions for LLM-based music AI. (i) LLM-as-controller/agent: systems such as AudioGPT and MusicAgent keep the LLM in charge of dialogue and planning while subtasks are dedicated to specialist audio models (e.g., ASR/TTS). (ii) LLM-as multimodal fusion/decoder: models like MuMu-LLaMA require a backbone with pretrained music/vision encoders and music decoders, which enables processing within one framework. (iii) Symbolic-only LLMs, which learn musical structure directly from compact score text (ABC). These excel at theory QA and structured composition.

The rest of this paper will proceed to expand in all three directions. First, this paper will synthesize three different LLM modelling approaches; next, review the experimental data that arises from said routes; then, an analysis including comparison, limitations and future prospects, before closing with an open outlook.

2. LLM modelling approaches

The three directions of LLM models, while different, all contribute vastly to the field of AI music synthesis and understanding by addressing complementary needs across composition, analysis, and production.

2.1. LLM as controller/agent

LLMs taking the form of a controller allows for the system to serve as a planner that parses user requests before decomposing them into subtasks. Each subtask is then routed to a special tool which processes it, such as Automatic Speech Recognition (ASR), source separation, transcription, tagging/captioning, and text-to-music synthesis. This design gives the system incredibly broad capability with minimal retraining, as well as being easy to extend by adding new tools. There have been several similar versions of such designs created in recent years, with the aforementioned AudioGPT being a favorable example of an LLM acting as a controller. Huang et al. utilized a four stage loop to develop such a system: modality transformation, which converts speech to text and back with ASR/TTS; task analysis that interprets the user's intent through the dialogue engine and prompt manager; model assignment, which selects the right pretrained audio model based on inputs like prosody, timbre, or language; and finally response generation, returning the final output to the user as text or audio. AudioGPT was able to produce consistent results when it came to processing audio information. The model consists of a library of speech/music/sound models and demonstrates

16 tasks spanning speech recognition/translation, sound event detection/extraction, text-to-speech/audio, image-to-audio, singing synthesis, and even talking-head generation. The agent used GPT-3.5 for reasoning and a lightweight GPU to host the tool models, displaying how audio functionality can be achieved by coordination rather than end-to-end retraining [1].

Similarly, Yu et al. proposed a music-focused agentic system MusicAgent, which falls in the same category. Its framework revolves around a tool registry with typed I/O that lets heterogeneous models/APIs work in conjunction. Comparable to AudioGPT, MusicAgent is also composed of multiple components: the task planner, tool selector, task executor, and response generator. A planner constructs a dependency graph in which lyrics are written first, followed by melody composition, and then arrangement and synthesis of audio. Upon receiving a request, the LLM analyzes it in detail and then "decomposes and organizes the requests into subtasks," where each subtask is carried out by a specialized tool. Finally, the LLM "subsequently organizes the output," allowing MusicAgent to function as a comprehensive and efficient music processing system [3].

Cocomposer is yet another system that incorporates LLMs for the same purpose. Five collaborating agents make up this multi-agent system, "each with a task based on the traditional music composition workflow" [4]. Xing et al. used three LLMs– GPT-4o, DeepSeek-V3-0324, and Gemini-2.5-Flash–to test their model. Cocomposer was evaluated in four categories: production quality, focusing on the technical aspects of the audio such as dynamics; production complexity, measuring how complex a generated audio is in terms of the number of components; content enjoyment based on how listeners viewed the piece; and content usefulness, determining how useful the generated music is as source material. The experiments concluded with Cocomposer being ruled as superior to single agent systems as well as "better interpretability and editability" than non-LLM music systems [4].

2.2. LLM as multimodal fusion

LLMs can also be used as multimodal fusion/decoder architectures for different modes and tasks. Fusion/decoder systems train lightweight adapters so an LLM can fuse encodings from multiple modalities (inputs) such as music, images and video. Afterwards, each kind of input is processed by a special encoder that extracts the respective features before being recombined in a way the model can interpret. Instead of picking external tools such as a controller agent, the model learns alignment between features and decoder controls, and thus results in one interface that supports understanding and creation. An example of this is MuMu-LLaMA, which builds on a dataset of roughly 167.7 hours of multi-modal data and uses pretrained models–in this case, MERT for music and ViT/ViViT for visual modalities–alongside decoders such as MusicGen and AudioLDM 2. The model was evaluated on four categories: Music understanding, Text-to-music generation, Prompt-based music editing, and Multi-modal generation. From the tests, Liu et al. concluded that the MuMu-LLaMA model outperformed those preceding it, as well as being able to produce music that matches prompts better than many baselines [2].

DeepResonance is another LLM designed for understanding music, "calibrated with multi-way instruction tuning on datasets that align music, text, images, and video, enabling it to learn from all four modalities jointly" [5]. Just like MuMu-LLaMA, it goes beyond simply music and text, bringing in images and video to help the model better understand its assignment. Mao et. al built new "Music4way" datasets (MI2T, MV2T, Any2T) that align music, visuals, and text, using them as data sets to train the model on. They also introduced ImageBind embeddings and a "pre-LLM fusion Transformer"–a type of model designed to take different kinds of data and embed them into a single shared space–to better fuse those different modalities before feeding into the system.

DeepResonance was then tested on six music understanding tasks using these datasets. The results displayed that LLM models which incorporated image and video tended to understand musical content better than those using just the standard music and text [5].

2.3. "Symbolic only" LLMs

Unlike the previous, "symbolic only" LLM approaches the score as language and is trained on sequences. The resulting outputs are mainly non-audible scores; however, the system can still produce audio later via other options such as a synthesizer or decoder.

ChatMusician is an example of a "symbolic" LLM over ABC (a music notation format), being an open source LLM trained on "4B-token music-language corpora MusicPile", a large dataset [6], and using 'MusicTheoryBench', an evaluation set containing 372 expert-curated, college-level questions to test music knowledge and reasoning. In all tests, ChatMuscician managed to beat LLaMA-2 and GPT-3.5 in zero-shot testing. Despite ChatGPT-4 still having a slight edge on knowledge, it still struggled on reasoning without prompt tricks compared to the model.

MuseCoco is another example of a 'text symbolic' LLM that turns given language prompts into editable MIDI/score formats. Designed by Lu et al., MuseCoco has around 1.2 billion parameters and is trained in a two stage format–a given text is translated to musical attributes and then to music [7]. This gives room for improved control and data efficiency as such attributes can be auto-extracted from MIDI for self-supervised training. In evaluations, MuseCoco reports gains over baselines in musicality, controllability, and overall score (≥ +1.27, +1.08, +1.32), and about +20% higher objective control accuracy [8].

3. Discussion

It is evident that all three directions for music LLMs were able to achieve a somewhat large degree of success, all fulfilling their designated tasks. All examples of controller agent-based LLMs were able to generate results which matched user requests: AudioGPT successfully accomplished routed user requests to specialist audio tools and adequately demonstrated multi-round handling of 16 tasks, while MusicAgent and CoComposer also automated request analysis and tool invocation over a large toolset, validating the method for potential music workflows. Agent-based LLMs also tended to benefit from multi-agent designs; this was demonstrated when CoComposer was compared to a similar single-agent model ComposerX: the former was able to outpace the latter in production complexity (PC), authenticating its superiority [4]. Meanwhile, the results from multimodal fusion/decoder systems demonstrate that a single backbone can align music with images, video and text, enabling both understanding and generation within one interface. Both MuMu-LLaMA DeepResonance excelled at their tasks of supervised captioning and QA over the MusicQA, MusicCaps, and Music4way-MusicCaps benchmarks. DeepResonance particularly achieved impressive results, surpassing MU-LLaMA– an older version of MuMu-LLaMA [9]– and every other baseline. The paper attributes the gains to a pre-LLM fusion stage plus ImageBind-based embeddings, which together allow the model to leverage cross-modal context more effectively than music–text-only models [5]. Finally, the experiments conducted on Symbolic LLMs show how this direction practicalizes knowledge. The zero-shot experiments conducted on ChatMusician's results on MusicTheoryBench were exceptional, proving that symbolic LLMs can be reliable for hitting specific structural targets and editing workflows without damaging the piece.

3.1. Challenges

Though every evaluated experiment accomplished its initial task, each still experienced complications when facing certain aspects. Agent-based LLMs encountered difficulties at the orchestration layer, the part of the system that plans and coordinates all the steps. This means long tool chains can amplify latency and propagate small upstream errors, so one failed ASR or separation call can derail the entire run. The results of AudioGPT, for example, display three key issues: its dependence on heavy prompt engineering, which is time-consuming; maximum token-length ceilings that could constrain multi-turn dialogue between the LLM and the user; and capability dependence hinging on the accuracy of the plugged-in audio foundation models [1]. CoComposer, as well, even with benefits from its multi-agent system, admits that while it outperforms in complexity and interoperability, non LLM based MusicLMs are able to generate better sounding audio. While multimodal fusion LLMS include backbones that reduce orchestration, they too have their own constraints. Liu et al. concluded that the reliability of MuMu-LLaMA on pretrained decoders negatively impacts quality; the system restricts controllability and long form structure due to a quality system that cannot be controlled [2]. Despite belonging in a different category, Chatmusician also had similar drawbacks. The current model was found to "mainly create Irish style music, as well as occasionally hallucinating; these issues were due to the data set on which it was trained and a lack of variety in given musical instructions [6]. Musecoco, despite outperforming GPT-4 and BART-base by a large margin, meets a roadblock when it comes to long sequences. However, Lu et al. proposed a potential use of the transformer Museformer to better manage long sequences [8].

3.2. Future prospects

Looking ahead, it's likely that LLMs for music will converge, and thus categorizing them will become increasingly unnecessary. Research on how office workers are using LLMs shows a shift from simple chatbots to more like smart assistants which assist daily workflows [10]. Music LLMs are no different and are likely to move into their own respective working niche as well.

As technology undoubtedly becomes more and more advanced, it is not unorthodox to hypothesize that the varying methods of Models, including more than just the ones discussed in this paper, will most likely aggregate to generate better results. Thus, LLMs of the future could have single hybrid systems that are able to achieve much more accuracy and quality than they can now. In the meantime, existing methods should seek to overcome any limitations. For controller agent systems, possible short term goals involve making a model faster, more accurate, and better suited for long form content. The latter is especially significant, as experiments conducted on AudioGPT found it to struggle with long form content. Selecting precise foundational models can reduce the risk of inaccuracy as well. As multi-agent orchestration already shows production quality and complexity benefits over single agents, further research in that direction is also key. For both multimodal fusions and symbolic score based LLMs, a key beneficial change could be using much more diverse pretraining data. For example, ChatMusician and MuMu-LLaMA were both found to be reliant on the dataset they were given, shown explicitly by the former always leaning towards the Irish Style music it was trained on. Adding variety would alleviate such problems, as it would reduce style bias and improve system generalization, leading to more faithful prompt following across genres. The earlier discussed use of transformers was suggested for models to better handle long form content.

4. Conclusion

Overall, this review traced three complementary paths in music LLMs and showed that each already delivers strong results at its core task, while also glossing over how each exposed fixable gaps in control, length, and data breadth. The paper sums up the three directions of LLM models: agent/controller systems, which use tool orchestration and multi-agent designs to route requests and reduce production complexity; multimodal fusion/decoder models that align music with text, images, and video to handle text-to-music, prompt-based editing, and image/video-to-music with strong listener preferences; and symbolic approaches which treats the score as language, excelling on musical foundation and improving controllability via a two-stage text to music format. In general, the limits noted are small next to the clear accomplishments. All systems already work well at their targets, and most issues are able to be mitigated with straightforward engineering and data work. The final outlook on this topic is pleasantly open and hopeful; the successful integration of music in Large Language Models opens many opportunities in the field of Artificial Intelligence engineering.

References

[1] Huang, R., Li, M., Yang, D., Shi, J., Chang, X., Ye, Z., ... & Watanabe, S. (2024). Audiogpt: Understanding and generating speech, music, sound, and talking head. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 21, pp. 23802-23804).

[2] Liu, S., Hussain, A. S., Wu, Q., Sun, C., & Shan, Y. (2024). Mumu-llama: Multi-modal music understanding and generation via large language models. arXiv preprint arXiv: 2412.06660, 3(5), 6.

[3] Yu, D., Song, K., Lu, P., He, T., Tan, X., Ye, W., ... & Bian, J. (2023). Musicagent: An ai agent for music understanding and generation with large language models. arXiv preprint arXiv: 2310.11954.

[4] Xing, P., Plaat, A., & van Stein, N. (2025). CoComposer: LLM Multi-agent Collaborative Music Composition. arXiv preprint arXiv: 2509.00132.

[5] Mao, Z., Zhao, M., Wu, Q., Wakaki, H., & Mitsufuji, Y. (2025). Deepresonance: Enhancing multimodal music understanding via music-centric multi-way instruction tuning. arXiv preprint arXiv: 2502.12623.

[6] Yuan, R., Lin, H., Wang, Y., Tian, Z., Wu, S., Shen, T., ... & Guo, Y. (2024). Chatmusician: Understanding and generating music intrinsically with llm. arXiv preprint arXiv: 2402.16153.

[7] Shu, Y., Xu, H., Zhou, Z., Hengel, A. V. D., & Liu, L. (2024). MuseBarControl: Enhancing fine-grained control in symbolic music generation through pre-training and counterfactual loss. arXiv preprint arXiv: 2407.04331.

[8] Lu, P., Xu, X., Kang, C., Yu, B., Xing, C., Tan, X., & Bian, J. (2023). Musecoco: Generating symbolic music from text. arXiv preprint arXiv: 2306.00110.

[9] Liu, S., Hussain, A. S., Sun, C., & Shan, Y. (2024). Music understanding llama: Advancing text-to-music generation with question answering and captioning. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 286-290). IEEE.

[10] Brachman, M., El-Ashry, A., Dugan, C., & Geyer, W. (2025). Current and Future Use of Large Language Models for Knowledge Work. arXiv preprint arXiv: 2503.16774.