MUSIC ORCHESTRATOR: MELODY-CONDITIONED MUSIC GENERATION FOR INSTRUMENT ALTERATION AND ENHANCEMENT

Sukru Samet Dindar, Siavash Shams
Department of Electrical Engineering
Columbia University, New York, US
{**sd3705**,**ss6928**}@columbia.edu

ABSTRACT

Recent advancements in AI-driven music generation have demonstrated the potential to transform the music production landscape. However, existing models often lack the precision and flexibility required for professional use, particularly in transforming single-instrument tracks into multi-instrument compositions without compromising the original artistic intent. This study introduces a novel framework that integrates textual and audio inputs to enable precise multi-instrumental transformations while retaining musical features such as melody, mood, and tempo. The model incorporates a Music Descriptor system, which synthesizes detailed technical descriptions—including tempo and key—alongside high-level attributes of input tracks, and a fine-tuned Music Orchestrator that uses these descriptions to generate coherent multi-instrument outputs. Evaluated against existing baseline models, our approach achieves superior performance in both alignment with user-defined prompts and the precision of the musical instrument classes, while keeping the melody-conditioning ability at a similar level. These results emphasize the potential of combining technical and high-level descriptors to improve AI music generation, bridging the gap between human creativity and machine-assisted production. Despite challenges such as dataset scarcity and instrument-specific inconsistencies, this work represents a significant step toward making AI tools more practical and accessible for both amateur and professional music producers.

1 INTRODUCTION

In recent years, the development of music generation models using generative AI and audio foundation models has advanced significantly Copet et al. (2023)Agostinelli et al. (2023). However, the quality of the music produced by these models remains a topic of skepticism. While AI-generated music has sparked controversy among both producers and consumers within the music industry, it is important to recognize that the music sector has historically embraced technological innovations. This can be seen in the emergence of new electronic instruments, the widespread adoption of music production software, and the dominance of digital streaming platforms. Therefore, it is important to identify a pathway for AI-assisted tools that supports music producers without dismissing technological progress or removing the human element from the production process entirely. When used appropriately, these tools have the potential to make music production more accessible, particularly for indie and amateur musicians. For instance, modern music production often involves the simultaneous operation of multiple instruments, a task that can be challenging for a single musician to execute at a professional level. Music generation models could be helpful in transforming a single-instrument track into a fully produced piece without compromising its original characteristics.

Despite the popularity of text-to-music (TTM) models, these tools often fall short in providing users with precise control over the output, limiting their application to experimental or novelty uses rather than as serious production tools. While text-based input allows for some customization such as specifying instruments, mood, or tempo, more detailed musical elements like intricate melodies are impractical to describe through text alone. To address this limitation, models must integrate audio input alongside text input, enabling users to influence the output more directly through reference

tracks. Although previous studies Copet et al. (2023) Wu et al. (2023) Mariani et al. (2024) have explored similar approaches, such functionality is typically treated as an auxiliary feature rather than the core focus of the model. Consequently, these models often yield suboptimal results, such as mismatches between the description prompts and the generated music or poor differentiation between instruments in the output.

This study introduces a novel model designed to take both a reference music track and a text prompt specifying the desired instruments for the output. The model generates music that faithfully adheres to the melodic and high-level features of the input, such as mood and tempo, while augmenting the track with the requested instruments. While the primary motivation of our model is to enable single-instrumental to multi-instrumental transformation, it also supports flexible configurations, allowing both input and output to be either single or multi-instrumental. Furthermore, we evaluate the model's performance using various metrics and benchmark it against baseline melody-conditioned text-to-music models. By addressing existing limitations and improving creative control, our model aims to bridge the gap between human creativity and AI-driven music production.

2 Related Works

Recent advances in AI-driven music generation have introduced various methodologies, using text and audio inputs to enhance creative control. Among the most prominent approaches, autoregressive language models have demonstrated significant potential in generating coherent and high-quality music. For instance, Agostinelli et al. (2023) introduced MusicLM, an autoregressive Transformer model capable of generating long-duration music with high fidelity from text descriptions. Similarly, Copet et al. (2023) proposed MusicGen, a single-stage autoregressive model that integrates both text and audio prompts, offering users greater flexibility in specifying desired musical attributes. These autoregressive methods emphasize sequence modeling, capturing temporal dependencies in music effectively. However, their reliance on text input limits fine-grained control over intricate musical details, causing the need for alternative approaches to address these challenges.

Another line of research employs diffusion-based models to overcome the limitations of autoregressive methods by generating music through iterative refinement. Schneider et al. (2023) developed Moûsai, a latent diffusion model designed to handle long-context music generation, enabling the creation of multi-minute tracks with stereo quality. This approach was further explored by Melechovsky et al. (2023), who introduced Mustango, a diffusion model tailored for controllable music generation based on detailed textual descriptions, including tempo, chord progressions, and mood specifications. Moreover, Lam et al. (2024) applied diffusion techniques to melody generation, using transformers to condition on chord progressions and generate rhythmically dynamic melodies. These diffusion-based methods excel in generating musically coherent outputs and offer enhanced flexibility, but they often require extensive computational resources, posing practical challenges for broader adoption.

In addition to autoregressive and diffusion-based models, other approaches have aimed to explicitly condition music generation on reference audio or lyrics. For example, Wu et al. (2023) explored the integration of audio inputs alongside textual prompts, treating this functionality as an auxiliary feature to improve user control. Meanwhile, Mariani et al. (2024) investigated the use of hybrid GAN models for generating melodies from lyrics, combining LSTM-based sub-networks to model long-term dependencies effectively. Similarly, Genchel et al. (2019) focused on explicitly conditioning melody generation with musically relevant features, demonstrating the benefits of incorporating domain-specific knowledge. Although these approaches offer promising results, they often lack the generalizability required for diverse music generation tasks, highlighting the need for models that seamlessly integrate multiple input modalities.

By building upon these methodologies, our work aims to unite the controllability and audio-text integration in a natural way, introducing a novel framework that enables precise multi-instrumental transformations while maintaining the creative intent of the input reference track. Our work specifically focuses on the instrumental changes while keeping the other high-level features of the reference music track similar, and it achieves to compose these multi-instrument tracks in a precise and harmonic way, a challenge that generic melody-conditioned music generation models struggle to overcome.

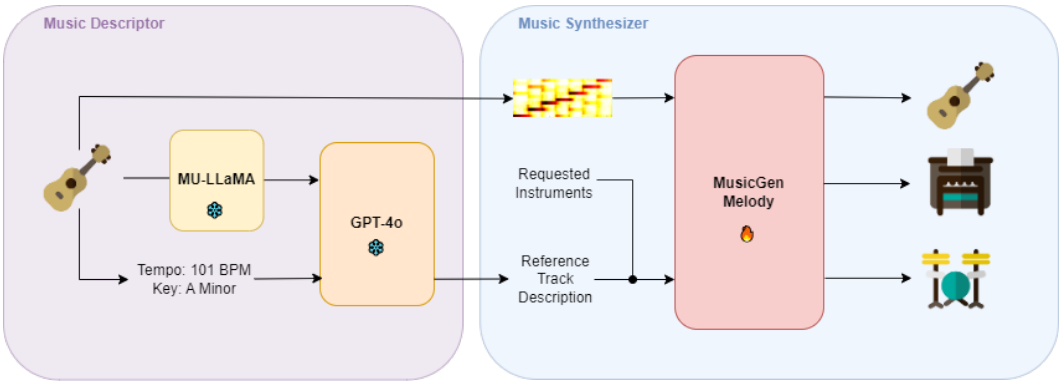

Figure 1: Overview of Music Orchestrator. The system consists of two parts. Music Descriptor (left) generates the detailed description of the reference track. Music Synthesizer (right) generates the output track conditioned on the description text and the reference track audio.

3 METHODOLOGY

The framework is composed of two main components. The first, the Music Descriptor, extracts a detailed and quantized description of the input music file. Initially, the reference track is processed by MU-LLaMA Liu et al. (2023) to obtain a high-level description. This description, along with technical musical attributes such as tempo and key, is then provided to GPT OpenAI (2024), which generates the overall description. The requested output instruments are subsequently combined with this description to create the final prompt. In the second component, the Music Synthesizer, the input audio is combined with the generated prompt to produce an output that incorporates the requested instruments while preserving the original musical properties of the input track.

3.1 MUSIC DESCRIPTOR

The first part of the model is a music description system taking the input audio provided as the input, and outputting the description of the music. While this part of the system is optional as the text input of the Music Synthesizer can be anything such as only the name of the requested instruments with the reference tone or even just empty input, the detailed description of the input music is expected to improve the general quality of the output music in addition to the similarity of some properties of input music and the output music. Since our model's main goal is composing the given relatively simple input music further with more instruments without changing the vibe significantly, a description generated from the input can be really useful to convey that information from input to output besides the spectral information provided by the chromagram input to the model.

For generating the description of a given track, there are some open-source music description models in the literature Liu et al. (2023) Doh et al. (2023). However, their main problem for our task is that this description is usually quite high-level, and more related to the mood and the feelings of the music rather than the technical details that describe the type of the music. While this expressions provide some direction, they are far from ideal for using these models as a music enhancement tool rather than a music generation tool from scratch.

For getting rid of this problem, there are two methods that can be used. One option is that already existing models can be fine-tuned using a more detailed description dataset. However, the biggest problem for this method is finding a suitable dataset for this task. The descriptions would require some music expertise from the annotators, and the copyright issues for the music makes it much harder for the curators. Because of this reason, the existing open-source music datasets in the internet is quite limited Agostinelli et al. (2023). Another option for including technical details into the music descriptions is using the external tools for getting these technical statistics, and later encorporate them with the high-level description provided by the music description models.

Due to the limitations of the first method, we used second method in our descriptor. Firstly, we obtain three different type of information pieces for the input music. First, we get the music description

using MU-LLaMA Liu et al. (2023) to get a general high-level description of the input track. Then, the tempo of the track is extracted from the MIDI file of the track provided in the dataset using pretty_midi package in python [1]. Lastly, the key of the track is analyzed using music21 package in python [2]. These three information sources are fed into GPT 4o language model OpenAI (2024) with the following prompt:

Create a short simple overall musical description with just a sentence using the following information. Do not include any instrument name. General description: {*HIGH-LEVEL DESCRIPTION*}*, Tempo:*{*TEMPO*}*, Key:*{*KEY*}

In this way, while the resulting description is pretty verbal, it still includes some technical information such as the tempo and the key of the track. This output is put together with the instruments requested for the output, and fed into the Music Synthesizer as the text input together with the audio input.

3.2 Music Synthesizer

The second part of the model carries out the main functionality of the model, and it is implemented for creating a model who can generate music according to the instructions while staying loyal to the melody and musical properties of the reference track. While this function can be used for turning a single-instrument track into a full song with multiple instruments, it is also possible to do other transformations such as single-instrument to single-instrument, multi-instrument to single-instrument or multi-instrument to multi-instrument.

For this part, called Music Synthesizer, the MusicGen Melody Copet et al. (2023) model is used as the baseline model. While structure-wise the model is similar to the motivation of Music Synthesizer, the pretrained model is not necessarily trained for a transition from single instrument to multi-instrument set-up. Thus, the output generated in the same way is usually not providing high-quality samples consistent with the provided prompt. Thus, we fine-tuned this model with a music separation dataset, Slakh-2100 Manilow et al. (2019), for teaching the model to capture the dynamics between instruments and learn how to compose multiple instruments together. For this goal, a new dataset is created by combining music single instruments and multiple instruments ranging from only 2 instruments to fully orchestrated mixes. In this way, the model learns to use the instruments together, rather than just translating the sound between different instruments.

The structure of the model is a single-stage autoregressive transformer decoder-based language model, conditioned on text and melody representation Copet et al. (2023). In the model, the input audio is tokenized by the audio tokenizer EnCodec Défossez et al. (2022) to obtain discrete tokens representing the audio. Then, a conditioning tensor is obtained from the input text and/or audio using transformers encoders. For melody conditioning , the input audio is first transformed into chromagrams, which map whole spectral audio information into one octave and create a suitable framework for representing the music. Then, dimentionality reduction and quantizing is applied to these chromagrams in order to condition the model together with the text tokens. The transformer model includes self-attention blocks and fully-connected blocks to process the inputs, and layer normalization is applied. Lastly, the output representations of the transformer model is turned into the audio sequences using linear layers.

4 Results

4.1 Dataset

For our dataset, we have selected Slakh2100 Manilow et al. (2019), a comprehensive collection consisting of 2,100 audio tracks and totaling 145 hours of mixed music, spanning 34 unique instruments including popular instruments for similar tasks such as guitar, piano, drum, bass, etc. One of the dataset's key strengths is its inclusion of isolated audio tracks for each instrument, which makes it particularly valuable for tasks that rely on conditioning the audio output based on individual instrument signals. This feature allows us to analyze and manipulate the contributions of

[1] https://github.com/craffel/pretty-midi

[2] https://github.com/cuthbertLab/music21

Table 1: Quantitative evaluation of the model with and without the music description

Method	SIM ↑	KL ↓	CLAP ↑
MusicGen Melody (w/o description)	0.615	1.380	0.148
Music Orchestrator (w/o description)	0.597	0.780	0.155
MusicGen Melody (w description)	**0.625**	1.467	0.208
Music Orchestrator (w description)	0.603	**0.766**	**0.214**

specific instruments within a full mix, offering a versatile foundation for experimentation in audio generation and conditioning-focused tasks.

For preparing the dataset, we used three different types of music pieces. Firstly, we included all the fully mixed tracks including every instrument available for that track to our dataset. Then, we added the single instrument stem tracks for model to capture the actual properties of all the instruments. Lastly, since the model might be dealing with just a few instruments for some tracks or some prompts, we also add extra submixes to the dataset for each track including 2, 3, and 4 instruments randomly chosen from the available instrument stem list of each track. This helps model to handle a range of instruments mixed in various numbers. It adds up totally to 25538 tracks in training set, 5375 tracks in validation set, and 2950 tracks in test set.

For creating the high-level descriptions later used for composing with other technical details, we used the music description model MU-LLaMA Liu et al. (2023). This model is obtained by fine tuning LLaMA-2 with 7B parameters, and the weights are published by the developers. For creating the final descriptions but synthesizing all information, the GPT API by OpenAI OpenAI (2024) is used. The prompts are given to the GPT 4o model, and the final descriptions are obtained. The whole process costs about $20.

4.2 Training

For the Music Synthesizer model, we used MusicGen Melody Copet et al. (2023) with their pre-trained weights. While there are two sizes of the model (medium with 1.5B parameters and large with 3.3B parameters), we used medium one due to the computational constraints.

For fine-tuning the model, Audiocraft [3] software published by Meta to use their audio generation models is used. While the details of the model are not fully explained in this paper, the corresponding information can be found in Copet et al. (2023). Although the original code is open-source, some modifications are made in the original Audiocraft code in order to adapt it to our problem. We fine-tune the model for 25 epochs with a scheduled learning rate and DADAM optimizer Nazari et al. (2019). It is observed that the error rate is plateaued after this fine-tuning, so we don't train the model further.

4.3 Evaluation

We evaluate the generated music using three objective metrics. The first is cosine similarity (SIM) between the reference track and the generated track. While this metric does not assess the model's ability to capture information specified in the description, it effectively measures the model's capacity to replicate the melody of the reference track. The second metric, KL divergence, benefits from a pre-trained audio classification model trained on AudioSet Gemmeke et al. (2017), as provided by the MusicGen codebase. KL divergence is calculated between the classification label probabilities of the original test set and the generated music. Lastly, the CLAP Elizalde et al. (2022) score measures the alignment between the textual description and the generated audio, capturing the model's performance in producing description-related outputs. A pre-trained CLAP model, also provided by the MusicGen codebase, is used for this measurement. For benchmarking, we use the original

[3] https://github.com/facebookresearch/audiocraft

pre-trained MusicGen Melody model as the baseline, as our Music Orchestrator model is fine-tuned on it.

To evaluate the impact of incorporating generated descriptions, we prepared two variations of the dataset. In the first variation, the Music Descriptor stage was omitted, and only the instrument names were provided as prompts to the Music Synthesizer. In the second variation, the generated descriptions were included as part of the prompts. Both the baseline model and our proposed model were tested on these datasets to observe the comparative analysis of their performance.

The results have revealed that while the melody-capturing ability of our model is slightly lower than that of the pre-trained MusicGen Melody, the cosine similarity score remains significantly higher than non-melody-conditioned models, which typically achieve only around 10% similarity. Hence, this slight decrease is negligible. For the other two metrics, the fine-tuned Music Orchestrator model outperforms all others. The KL divergence metric shows significant improvement with fine-tuning, demonstrating the model's enhanced ability to incorporate given instruments into the generated music. Similarly, the CLAP metric indicates that the inclusion of generated descriptions greatly improves performance for both the baseline model and the Music Orchestrator. This emphasizes the critical contribution of these descriptions to the model's overall performance.

5 Discussion

Our work demonstrates both the potential and current limitations of AI-powered music orchestration systems. While audio models have seen significant advancement and widespread adoption, they remain far from being practical tools for professional music production. The primary challenge lies in the gap between generating music from simple text prompts and providing the fine-grained technical control that musicians require. Current models excel at direct text-to-music generation but struggle with precise instrumental arrangements and specific musical modifications, limiting their utility in professional settings. Additionally, the handling of different instruments varies significantly in quality - while sustained instruments like strings and piano generally produce consistent results, percussion instruments present unique challenges due to their sparse and rhythmic nature.

The scarcity of comprehensive music datasets remains a significant bottleneck, primarily due to copyright restrictions in the music industry. This limitation affects not only the training of models but also their ability to learn diverse musical styles and arrangements. Our experiments revealed that incorporating technical details (tempo, key) alongside high-level descriptions markedly improves the model's performance, as evidenced by the improved CLAP scores. Looking forward, several key improvements could enhance the system's capabilities: developing more sophisticated music description models, better handling of instrument-specific characteristics (particularly for percussion and other sparse instruments), and finding innovative solutions to the dataset limitation problem. These advancements would be crucial steps toward making AI music tools more practical for professional music production while maintaining the creative control that artists require.

6 Conclusion

This work presents a novel framework for precise multi-instrumental music generation, combining textual and audio inputs to enhance creative control. By integrating detailed musical descriptions, our model addresses key limitations in current systems, achieving improved alignment with user-defined prompts. While challenges like dataset scarcity and instrument-specific inconsistencies persist, our approach highlights the potential of AI to augment music production, bridging the gap between technology and artistic creativity.

REFERENCES

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating music from text, January 2023.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, June 2023.

SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. Lp-musiccaps: Llm-based pseudo music captioning, 2023. URL https://arxiv.org/abs/2307.16372.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression, 2022. URL https://arxiv.org/abs/2210.13438.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision, 2022. URL https://arxiv.org/abs/2206.04769.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Benjamin Genchel, Ashis Pati, and Alexander Lerch. Explicitly conditioned melody generation: A case study with interdependent rnns. *arXiv preprint arXiv:1907.05208*, 2019.

Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, and Ying Shan. Music understanding llama: Advancing text-to-music generation with question answering and captioning, 2023. URL https://arxiv.org/abs/2308.11276.

Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity, 2019. URL https://arxiv.org/abs/1909.08494.

Giorgio Mariani, Irene Tallini, Emilian Postolache, Michele Mancusi, Luca Cosmo, and Emanuele Rodolà. Multi-source diffusion models for simultaneous music generation and separation, 2024. URL https://arxiv.org/abs/2302.02257.

Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. *arXiv preprint arXiv:2311.08355*, 2023.

Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization, 2019. URL https://arxiv.org/abs/1901.09109.

OpenAI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o.

Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Mo\^ usai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan. Music controlnet: Multiple time-varying controls for music generation, 2023. URL https://arxiv.org/abs/2311.07069.