Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ23840

Composing Music Based on Advanced Scenarios: Autoregressive Models and Diffusion Models

Wenzhe Lin

North London Collegiate School Dubai, Dubai, China
wenzhelin60820@gmail.com

Abstract: As a matter of fact, music composition is of the hot topics in contemporary society. With the proposal of the state-of-art models, great progress has been achieved in terms of music composing based on autoregressive models as well as diffusion models. In this paper, the benefits and drawbacks of autoregressive models and diffusion models in the context of AI music generation are compared. To be specific, this study places its focus on 4 specific models, i.e., CARGAN, SaShiMi, FluxMusic, and MeLoDy. According to the analysis, AR models generate music sequentially, synthesizing high quality music, but suffers from issues like poor coherence over longer pieces and low efficiency. In contrast, diffusion models iterate over the entire waveform to refine details, which leads to better overall structure but can be much more resource intensive. Nevertheless, both methods have promising prospects, and a hybrid approach could have great future potential. Overall, these results offer a guideline for further exploration of music composition.

Keywords: Composing music, CARGAN, SaShiMi, FluxMusic, MeLoDy.

1. Introduction

In the middle of the 20th Century, British computer scientist Alan Turing developed a system for early computers to generate musical notes, which was believed to be the earliest instance computer generated musical sound [1]. This laid the foundation for the field of computer generated music, starting with computers generating individual notes and sounds to be used by music producers. Electronic music, as a genre, is a great example of this, as it makes use of computer instruments - an idea developed further by contemporary artists [2]. Other ways computers have been used in the process of music production include algorithmic generation of specific waveforms (NyquistIDE), synthesis of human voice (Vocaloid), and digital music formats (MIDI) [3].

A significant difference that separates modern AI-generated music from early computer music is that, while early utilizations of computers for musical applications are mainly computer instruments and synthesis of individual notes or musical components, new AI are able to generate, analyze, and improve entire pieces of music [4]. One of the most well-known AI music generation models is MusicLM by Google [5]. This model is capable of synthesizing high-quality music from a variety of text prompts, which allows the users to specify characteristics of the desired result, including genre, style, emotion, and instruments. The model has access to a vast dataset of existing music, which gives it the ability to produce music that sound, at times, indistinguishable to pieces created by

DOI: 10.54254/2755-2721/2025.TJ23840

humans. Another notable model is AIVA (Artificial Intelligence Virtual Artist), which has been popular in professional settings that require its specialty in classical and orchestral music [6]. Some industries that commonly use AIVA include film and video games, since this model is highly effective at generating emotional and coherent tracks. Suno AI, another emerging AI model, focuses on making AI music generation more accessible for all people, at the benefit of smaller studios or independent creators [7].

As AI-generated music continues to develop, its applications have expanded beyond just composition. Many artists and producers now use AI as a creative assistant, helping with tasks such as generating melodies, harmonies, and even lyrics. This has allowed for more experimentation in music production, but it has also raised ethical concerns regarding originality and copyright. Since these models are trained on existing music, questions arise about whether AI-generated tracks can be considered truly original, or if they are simply advanced recombination of past works. These issues remain a topic of debate among musicians, AI developers, and legal experts. AI music, as a field of study, is becoming increasingly prevalent in the modern day. This is because of the convenience and utility it offers to composers and producers, while still maintaining a high quality for the final piece. However, despite its presence in many professional settings, it is still an area with much to explore, in terms of musical styles, technical approaches, and future practical applications.

This research will have its focus on two specific AI music models, autoregressive models and diffusion models. Each model represents different approaches in the way AI music is synthesized. Autoregressive models operate by a note-by-note generation method, similar to traditional music composition. Diffusion models generate from a random noise, then iterating to refine the entire piece until a final track is produced. This research will compare the advantages and drawbacks of each model, providing some analysis on the future prospects of the models.

2. Description of models

Simple autoregressive models were some of the earliest approaches to computer music [8]. In recent years, it has developed and transformed drastically, becoming much more advanced. In essence, autoregressive (AR) models generate music sequentially, or note by note. These models predict each new note or element by analyzing previous elements using an underlying algorithm. AR models commonly use architecture such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks. They are capable of processing temporal dependencies but fail at musical coherence when generating longer pieces. Modern AR models may use transformer-based methods to improve coherence and structure for longer works. These models are similar to large language models, popular today, in how they generate music. However, a significant flaw of this approach is that this process may introduce errors that are carried forward, affecting a large portion of the final music. The fact that AR models generate sound discreetly, or element by element, makes it great for music formats like MIDI or sheet music. At the same time, it requires converting the synthesized piece into raw audio or waveforms, which can be computationally demanding.

In contrast, diffusion models synthesize music through an iterative process [9]. Starting with noise, these models refine the noise and transform it into music by creating structure and details like melodies or harmonies. Unlike AR models, diffusion models directly manipulate random sound to make it sound more musical, which may require the use of spectrograms or latent spaces to retain features important to music like time. Therefore, diffusion models are often better at musical structure, using ideas important in music theory such as leitmotifs or structure specific to a genre. Overall, diffusion models have better performance when producing entire pieces, such as text-to-music generation, as it does not have issues like error propagation. However, diffusion models are

DOI: 10.54254/2755-2721/2025.TJ23840

highly computationally intensive and are often slower, as they require up to thousands of steps or iterations in the sampling stage. Diffusion models also tend to require large training datasets to be effective.

In the future, approaches or architectures that incorporate both AR and diffusion methods may offer more holistic solutions that do not have significant drawbacks in most situations. However, as of present, both models show potential in future applications as they have their own strengths that makes one more favorable than the other in certain circumstances.

3. Autoregressive models

This section will detail the capabilities of AR music generation models, using the examples of 2 models, CARGAN and SaShiMi.

3.1. CARGAN

CARGAN, a model developed by Morrison et al for the purposes of generating waveforms from mel-spectrograms, is an instance of modern AR models [10]. The model follows the AR structure while integrating methods of generative adversarial networks (GANs) to address limitations faced by traditional GAN models. Crucially, CARGAN consists of 3 main components. Firstly, an AR conditioning stack which summarizes the previous samples into a vector through its complex algorithm. Secondly, a generator network that takes input conditioning and AR conditioning and converts them into a waveform. The final component is a series of discriminators (a vital part of the GAN approach) that provide adversarial feedback. One major improvement CARGAN makes compared to traditional AR models is that it generates audio in chunks instead of one sample or element at a time. This is important because it makes the model much more efficient while maintaining the advantages of AR models.

As shown in the Table 1, in 5 out of 6 metrics measured, CARGAN performed best between all the models tested. CARGAN also performs well when it comes to audio quality, with relatively few artifacts and disruptions. Another positive is that CARGAN is trained quickly at high efficiencies compared to other GAN models, reducing training time by 58%. Additionally, it also lowers pitch errors by 60%, in comparison to prior state-of-the-art GAN models. However, the chunk based approach does introduce some limitations. For example, the chunks can have discontinuities or artifacts on their boundaries, which requires additional processing to address. The training dataset CARGAN uses is biased in the genres of music included and their proportions, leading to lower performances when synthesizing some genres of music.

Table 1: Objective evaluation results for spectrogram-to-waveform inversion on VCTK and DAPS

Method	VCTK			DAPS		
	Pitch↓	Periodicity↓	F1↑	Pitch↓	Periodicity↓	F1↑
HiFi-GAN	51.2	.113	.941	54.7	.142	.942
CARGAN	29.4	.086	.956	21.6	.107	.959
- GAN-TTS	37.9	.099	.949	27.0	.117	.953

DOI: 10.54254/2755-2721/2025.TJ23840

- Loss balance	33.7	.104	.943	34.1	.119	.952
- Prepend	24.6	.088	.955	24.4	.108	.958

3.2. SaShiMi

SaShiMi, developed by Goel et al., is an AR music generation model that builds upon the S4 framework [11]. The model consists of 2 main components. Firstly, the model leverages the speed of the S4 architecture, as it is highly efficient when working with raw waveforms. SaShiMi uses parameterisation to make the S4 algorithm stable, which is important to S4's performance. This is an unusual approach compared to the other AR models, which typically use transformers or CNNs. An innovation SaShiMi makes is that it applies S4 layers to raw audio generation. Each layer consists of S4 blocks. The effect of this is that the model generates in multiple tiers at different temporal resolutions - that is to say there are coarser layers and finer layers, which are combined to generate a final waveform. Lower tiers can help outline structure and rhythm, while higher layers can provide fine details like texture. Interestingly, SaShiMi is able to generate both autoregressively and non-autoregressively. In AR mode, the model uses probabilistic equations to determine sequential outputs based on prior information. In this mode, it has issues with stability, so SaShiMi is constrained to Hurwitz matrices, which are also called stable matrices, when generating autoregressively.

In the Table 2, it is evident that SaShiMi is capable of high performance, as it does better than the baseline models used by Goel et al. One of SaShiMi's advantages is that S4 layers are great at handling long sequences, and can efficiently synthesise high-resolution audio. Another strength of SaShiMi is thanks to the use of Hurwitz matrices - it ensures stability throughout the entire waveform, preventing degradation. However, despite its strengths, SaShiMi is limited by the fact that it requires Hurwitz matrices in AR mode, meaning that it is less flexible than some other models, like WaveNet. Another drawback of SaShiMi is that in the training stage, while efficient, it is resource intensive, limiting the use cases of the model. Lastly, the AR model, according to Goel et al., is inherently slow and inefficient compared to diffusion models.

Table 2: Results on AR modeling of Beethoven, a benchmark task, where SaShiMi outperforms all baselines while training faster

Model	Context	NLL	@200K steps	@10 hours
SampleRNN*	1024	1.076	-	-
WaveNet*	4092	1.464	-	-
SampleRNN†	1024	.099	1.125	1.125
WaveNet†	4092	.104	1.088	1.352
SaShiMi	128000	0.946	1.007	1.095

4. Diffusion models

DOI: 10.54254/2755-2721/2025.TJ23840

To evaluate the capabilities of diffusion models, this section will focus on 2 specific models, FluxMusic and MeLoDy.

4.1. FluxMusic

FluxMusic is an AI music generation model developed by Fei et al [12]. This model is built upon the FLUX text-to-image AI model - producing an AI music generation model with a transformer-based architecture. FluxMusic uses pre-trained text encoders to extract information from the input prompt and derive “appropriate representations”. This will result in a more accurate interpretation of the user prompt by the model, generating music that better adheres to the input. FluxMusic uses multiple text encoders, this will provide a more flexible approach that allows the model to work with different types of user input prompts.

An important feature of FluxMusic is that it represents music as mel-spectrograms. Each 10.24 second segment of audio is converted into a 64×1024 mel-spectrogram. This representation is then, using a Variational Autoencoder, compressed into a 16×128 latent representation. This latent space is used as the base for noise addition and model training. At the end of music generation, a pre-trained speech generation model that runs on mel-spectrograms is used to reconstruct the waveform.

The FluxMusic model also builds upon the Flux architecture, as shown in Fig. 1, starting by constructing an input that consists of text and noise. After, the model uses a double stream block and a single stream block. In the double stream block, the model uses two distinct sets of weights for both the text and music components. The two sets are treated as independent transformers that merge during the attention phase, considering the other transformer while maintaining their independence. In the single stream block, the model focuses solely on the musical component. The double stream and single stream blocks iterate over the latent space to convert noise into music.

Overall, the model uses the diffusion approach through the double stream block and single stream block transformers, starting with random noise then iterating to refine its details. FluxMusic has advantages such as being able to use a variety of input prompts, not limited to a specific style. The fact that FluxMusic uses latent spaces and mel-spectrograms to process sound makes it more efficient, while keeping important properties for music generation. In the Table 3, it is also evident that FluxMusic has clear advantages over many other models, requiring much less training data while still having strong results in many different criteria. However, FluxMusic faces challenges when working with certain writing styles, even with multiple encoders. The model may struggle with human nuance in the prompt. The paper is also somewhat limited in evaluations, so FluxMusic’s performance in some areas were underexplored/not explored.

DOI: 10.54254/2755-2721/2025.TJ23840

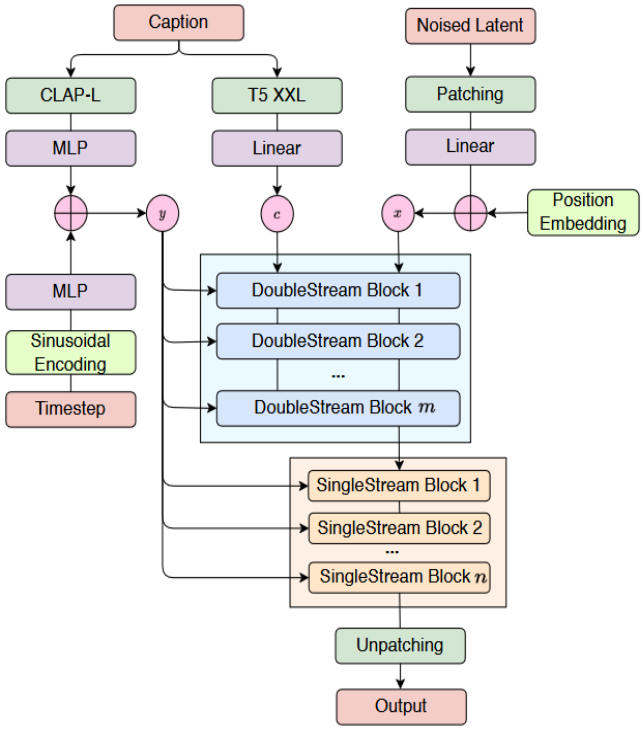

Figure 1: A diagram demonstrating the infrastructure of FluxMusic [3]

Table 3: Evaluation results for text-to-music generation with diffusion-based models and language-based models

Model	Params	Hours	FAD ↓	KL ↓	IS ↑	CLAP ↑	FAD ↓	KL ↓	IS ↑	CLAP ↑
MusicLM	1290M	280k	4.00	-	-	-	-	-	-	-
MusicGen	1.5B	20k	3.80	1.22	-	0.31	5.38	1.01	1.92	0.18
Mousai	1042M	2.5k	7.50	1.59	-	0.23	-	-	-	-
Jen-1	746M	5.0k	2.0	1.29	-	0.33	-	-	-	-
AudioLDM 2 (Full)	712M	17.9k	3.13	1.20	-	-	-	-	-	-
AudioLDM 2 (Music)	712M	10.8k	4.04	1.46	2.67	0.34	2.77	0.84	1.91	0.28

DOI: 10.54254/2755-2721/2025.TJ23840

QA-MDT (U-Net)	1.0B	12.5k	2.03	1.51	2.41	0.33	1.01	0.83	1.92	0.30
QA-MDT (DiT)	675M	12.5k	1.65	1.31	2.80	0.35	1.04	0.83	1.94	0.32
FluxMusic	2.1B	22K	1.43	1.25	2.98	0.36	1.01	0.83	2.03	0.35

4.2. MeLoDy

MeLoDy is a model created by Lam et al., with the goal of generating music efficiently from text prompt [13]. The model uses an LM-guided diffusion approach to generate music. Similar to FluxMusic, MeLoDy transforms text prompts and audio inputs into latent space representations, and decodes these representations using a Variational Autoencoder. One of the key techniques MeLoDy uses is a process named Dual-Path Diffusion (or DPD). In essence, DPD uses Coarse-Path Processing and Fine-Path Processing to generate music with high efficiency. In the Coarse-Path Processing phase, the model iterates over the initial noise to create a low-resolution foundation for the music, which determines the overall structure and genre. In the Fine-Path Processing stage, MeLoDy refines the details of the audio, such as melody and harmony, which generates at high-resolution.

In summary, MeLoDy takes an LM-guided diffusion approach and DPD to generate music from noise. The model is notable for its high efficiency and adherence to user prompts. Lam et al. used the following chart to demonstrate the capabilities of MeLoDy. Their model is able to:

•AC:being able to continue a piece of music given

•FR: being able to sample at a higher rate than real time (with a V100 GPU)

•VT: having been tested using a variety of prompts, including instruments, genres, and long-form descriptions

•MP: whether the evaluations were performed by music producers

From the Table 4, it can be seen how with similar sizes of datasets, MeLoDy is far more efficient than Google's MusicLM - according to its creators, MeLoDy reduces "95.7% or 99.6% forward passes in MusicLM, respectively, for sampling 10s or 30s music." Another advantage of MeLoDy is that music producers find it to perform more favorably in audio quality, while other models such as MusicLM and Noise2Music are stronger at musicality and text correlation. Compared to other models, MeLoDy is also notably more scalable, meaning that it can handle longer text prompts and audio - MeLoDy is also capable of generating uninterrupted, lengthy pieces of music, unlike MusicLM. Overall, MeLoDy has strong potential in some use cases that may require models to generate in real-time. However, MeLoDy is not without limitations. From the paper, one of the most significant limitations is that it is unable to generate music with vocals effectively, reducing its use-cases and range of effective prompts. This is due to the training dataset having few pieces with vocals, in an effort to prevent disruptions caused by "unnaturally sound vocals". Furthermore, the training data used was somewhat unbalanced and had been slightly biased towards pop and classical music, which may have an effect on the music MeLoDy generates. Finally, the dynamics of a long generation may be limited as a result of the training conducted.

Table 4: Evaluation results for text-to-music generation with diffusion-based models and language-based models

DOI: 10.54254/2755-2721/2025.TJ23840

Model	Prompts	Training Data	AC	FR	VT	MP
Moûsai	Text	25k hours of music	✓	✓	✗	✗
MusicLM	Text, Melody	280k hours of music	✓	✗	✓	✗
Noise2Music	Text	340k hours of music	✗	✗	✓	✗
MeLoDy	Text, Audio	257k hours of music	✓	✓	✓	✓

5. Comparison, limitations, and prospects

Through comparing the models of CARGAN, SaShiMi, FluxMusic, and MeLoDy, overall advantages and limitations of AR and diffusion approaches can be seen. AR models like CARGAN and SaShiMi generate waveforms with higher quality and stability in comparison to other popular models like WaveNet. Moreover, the AR approach typically has lower training times and is less computationally intensive due to having lower mean iteration steps. AR models also perform well when it comes to fine details, like individual notes or elements in comparison to other approaches. However, AR models suffer in long-term coherence and overall musical structure, as they generate sequentially. Another common drawback is slow inference speeds, which may make these models unviable for real-time applications. These downsides make AR models less versatile and impact their appropriateness for many applications.

On the other hand, diffusion models excel at long-term structure and coherence, due to their nature of iterating over the whole waveform. These models, like FluxMusic and MeLoDy, can maintain forms specific to genres and use musical techniques like leitmotifs. Despite having more iteration steps, some diffusion models, like MeLoDy, are more efficient and can be used in real-time. However, diffusion models may struggle at finer details, synthesising inaccurate notes. Due to the diffusion approach of generating from random noise, it is also common for diffusion models to have artifacts and noise disruptions, which can also be caused by having vocals in a model's training data. Finally, a significant drawback of diffusion models is that they are often highly computationally intensive, which can limit their usage for smaller projects/studios, or independent producers.

Overall, both approaches, AR and diffusion, have promising future prospects, each having their own application settings. AR models are better suited to smaller projects and individual producers as they generally use less resources, while diffusion models may be used in more professional settings and large industries due to these projects being able to satisfy their computational requirements, and in return, gaining advantages like real-time generation or more holistic musical structure. A hybrid approach, if done correctly, may be able to combine the benefits of both models and minimize the drawbacks. This hybrid AR diffusion approach could be a development that has a permanent impact on computer music, especially AI music generation.

6. Conclusion

To sum up, this research discusses AI music composition/generation through comparing two models, the AR model and the diffusion model. Specific AI models are referenced, taking into consideration their mechanisms, applications, as well as advantages and limitations. AR models, such as SaShiMi and CARGAN have the advantages of producing high quality waveforms with few artifacts. Additionally, lower training times is also a common benefit for AR models. However, they

DOI: 10.54254/2755-2721/2025.TJ23840

may suffer from issues with efficiency, which limits their uses and versatility. As for diffusion models, FluxMusic and MeLoDy are notable in their specific advantages, being more effective and efficient in some cases compared to established models like Google's MusicLM. Through analyzing these examples, it can be concluded that both AR models and diffusion models have strong, promising potential for future musical applications. This paper set out to explore the advantages and drawbacks of the two types of AI music generation models, as well as to speculate or give an opinion on their future prospects.

References

[1]Copeland, B.J. and Long, J. (2017) Turing and the History of Computer Music. Boston Studies in the Philosophy and History of Science, 189–218.

[2]Chapter Six: History of Electronic and Computer Music. (n.d.). Cmtext.indiana.edu. Retrieved from https://cmtext.indiana.edu/history/chapter6_timeline.php

[3]Lee, E. (2024) AI and the Sound of Music. Yalelawjournal.org. Retrieved from https://www.yalelawjournal.org/forum/ai-and-the-sound-of-music

[4]Spacefood, A. K. (2019) The Singing Synthesizer That Can Make Realistic Computer Vocals [Yamaha Vocaloid 5 Plugin Tutorial & In-Depth Review]. Warp Academy.

[5]MusicLM - AI Model for Music Generation. (n.d.). Retrieved from Musiclm.com. https://musiclm.com/

[6]Lauder, E. (2017). Aiva is the first AI to Officially be Recognised as a Composer | AI Business. Aibusiness.com. https://aibusiness.com/verticals/aiva-is-the-first-ai-to-officially-be-recognised-as-a-composer

[7]Francis, A. (2024, December). Suno's Latest AI Music Generator: 5 Reasons It's a Game-Changer. EWEEK. Retrieved from https://www.eweek.com/news/suno-latest-ai-music-generator/

[8]What are the key differences between autoregressive models, latent variable models, and implicit models like GANs in the context of generative modeling? (2024, June 11). EITCA; EITCA. Retrieved from https://eitca.org/artificial-intelligence/eitc-ai-adl-advanced-deep-learning/advanced-generative-models/modern-latent-variable-models/examination-review-modern-latent-variable-models/what-are-the-key-differences-between-autoregressive-models-latent-variable-models-and-implicit-models-like-gans-in-the-context-of-generative-modeling/

[9]GANs vs. Diffusion Models: In-Depth Comparison and Analysis. (2024). Sapien.io. Retrieved from https://www.sapien.io/blog/gans-vs-diffusion-models-a-comparative-analysis

[10]Morrison, M., Kumar, R., Kumar, K., Seetharaman, P., Courville, A. and Bengio, Y. (2021). Chunked Autoregressive GAN for Conditional Waveform Synthesis. ArXiv, arxiv.2110.10139

[11]Goel, K., Gu, A., Donahue, C. and Ré, C. (2022) It's Raw! Audio Generation with State-Space Models. ArXiv, arxiv.2202.09729

[12]Fei, Z., Fan, M., Yu, C. and Huang, J. (2024). FLUX that Plays Music. ArXiv, arrxiv.2409.00587

[13]Lam, M.W.Y., Tian, Q., Li, T., Yin, Z., Feng, S., Tu, M., Ji, Y., Xia, R., Ma, M., Song, X., Chen, J., Wang, Y. and Wang, Y. (2023) Efficient Neural Music Generation. ArXiv, arXiv.2305.15719