JOURNAL of COMPUTER SCIENCES INSTITUTE

JCSI 38 (2026) 66-72
Received: 23 September 2025
Accepted: 19 December 2025

Analysis of the impact of machine learning algorithms on the quality of generated sounds

Krzysztof Pedrycz, Mateusz Pikula*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Music generation using broadly understood AI is an evolving field with many challenges and opportunities. This thesis explores the use of generative adversarial networks for this endeavour, focusing on and comparing variety of different solutions that are already developed. Various architectures were tested and evaluated, in order to find the most effective approach to generating music. The results demonstrate that, although there are many solutions that can generate music that is both coherent and creative, there is still place for improvement in terms of model stability and created music quality. This work contributes to the understanding of generative adversarial networks in music generation and provides a foundation for future research in this area.

Keywords: GAN; music; generation

*Corresponding author

Email address: s95525@edu.pollub.pl (M. Pikula)

Published under Creative Common License (CC BY 4.0 Int.)

1. Introduction

Music is one of the most elusive and complex forms of art. It is a universal language that transcends cultural and linguistic barriers while also being deeply personal and subjective. Generating music that can be considered on par with its human-made counterparts is a task that has driven many programmers and musicians alike to create their unique solutions to said problem. This study aims to compare a few of the most notable ones based on generative adversarial networks (GANs) [1], how they managed such tasks, and what issues they encountered. In doing so, our aim is to create a background for future tests and breakthroughs in this field. Firstly, a closer look was taken at GANs in general, their inner workings, many different versions, and applications. It is mandatory to understand how they operate and how GANs evolved after their introduction in 2014 to be able to compare specific models in specific fields like music generation. The next topics addressed in this paper are more music centered. Matters like differences between specific file formats used in the music industry or approaches to music generation itself. After that, having established the technical properties of the environments where the tests were run, said tests were conducted. Then, the tests' results were evaluated using a plethora of metrics in an attempt to objectively assess and rate each of the models that participated in the testing.

2. Literature review

Nowadays, the rapid development of AI provides us with not only new ways of solving various problems but also new, interesting approaches to creating art. GANs are neural networks made with the aim of generating realistic data. They have found widespread usage in producing both pictures and videos, with tools such as Midjourney, DeepAI or Sora becoming extremely popular in recent years.

However, the task of generating music in similar ways is much more demanding and challenging. Music is usually composed of multiple instruments and vocals closely interacting with each other, making it much more demanding to recreate by AI [2]. Attempts were made to create GAN models tailored to such tasks with various effects. This work aims to investigate and compare a few of the most promising of them.

2.1. What is GAN?

Generative Adversarial Network (GAN) is a framework for estimating generative models proposed by Ian Goodfellow in his 2014 paper [1] Said framework consists of two separate parts:

- generative model (later as generator) — model that uses labeled training dataset to learn differences between provided data, used for supervised classification,
- discriminative model (later as discriminator) — model that is often used with unlabeled training dataset to generate new data that is similar to one provided,

They are competing with each other in the form of a zero-sum game; the generator generates new data, while the discriminator assesses whether the sample is taken from the dataset or made by the generator. The sole training criterion is competition between those two models, where the generator tries to create a sample that the discriminator determines to be coming from the dataset. If both the generator and discriminator are multilayer perceptrons (MLPs), GAN can be trained with backpropagation.

Consequently, after GANs' introduction, there were many new versions developed with alternative architectures, such as Conditional GAN [3], Deep Convolutional GAN [4], Laplacian Pyramid GAN [5], Cycle-Consistent

GAN [6], Self-attention GAN [7] or Deep Regret Analytic GAN [8].

There are many ways to create a GAN model, but several frameworks provide tools and libraries that greatly enhance and simplify the process. As described in [9], the most popular are TensorFlow, PyTorch and Keras.

2.2. Applications of GANs

Being a versatile and powerful tool, that GAN is, it has found many applications in various fields; first and foremost, the obvious application of GANs is generating images, videos and 3D objects [10] [11] [12], as this is what the majority of research is focused on. It is, though not the only application of this technology, as it can also be used to generate datasets [13] [14], reconstruct photos and enhance their resolution [15] [16] [17] or help improve healthcare by creating automatic disease detection systems [18].

Raising the subject of GANs applications, we cannot in good conscience omit the fact of their potential malicious use. It can be used to create fake identities or, for example, fabricate false medical records [19].

2.3. Music generation approaches

Music generation may seem like a straightforward task similar to image generation, but in reality, it is much more complex. Music is a form of art that is based on many factors, such as rhythm, melody, harmony, timbre, dynamics, and lyrics. All of those elements have to be taken into consideration when creating a model in order to achieve satisfying results. In this subsection, a closer look will be taken at some of the most interesting works in the field of music generation using GANs.

2.4. Difference between music and sound file formats

File formats are a crucial part of music generation, as they determine the way the music is stored, processed, and how many details are included. There are many different formats, serving different purposes, but for this topic, 2 are the most important: MIDI and WAV.

MIDI, or Musical Instrument Digital Interface, is not a standard file format but more of a data communication protocol. As described in [20],” The difference between audio information and MIDI data is similar to the difference between a tape recording of a pianist performing a piano sonata and the sheer music for that same sonata. The recording captures the musical information itself, storing the actual sounds that came from the piano. Data, on the other hand, represents information for the purposes of storing or transmitting it in a more compact or more easily transmitted form”.

Thus, MIDI is not used to store actual sounds, like WAV, MP3 or FLAC, but rather raw information about the sound. This approach provides both limitations and advantages. MIDI is broadly used in the creation of digital or electronic music, as it allows for tremendous flexibility in terms of editing and control over the track or even single notes. Its main advantage in the field of music generation is its richness in information about the music piece, as it contains not only notes but also information about pitch, rhythm, tempo, repetition of single notes, and much more. This makes it a perfect choice for music generation, as it provides a lot of data to work with. However, MIDI also has limitations. For example, MIDI does not specify any particular sounds that an instrument must make. Thus, it does not help in keeping the quality and range of created pieces in any way. There are also technical shortcomings. The most prevalent one is that the MIDI format contains only 16 channels of data. This might not be enough for any music pieces containing many instruments and vocal parts.

WAV, originally named WAVE for Waveform Audio File Format, on the other hand, is a standard for audio data. Presented by IBM and Microsoft back in 1991 [21], WAV has become a staple in the music industry, as it is uncompressed, hi-res, and LPCM (Linear Pulse-Code Modulation) using audio format.

2.5. Music generation using GANs

The first GAN model prepared specifically for music synthesis that is important to mention is multi-track sequential GAN (MuseGAN), proposed by Dong et al. [2], as the authors tried to tackle the problem that most prior studies chose to simplify. Previous attempts settled for generating only single-track monophonic music or polyphonic music as a combination of several monophonic melodies. MuseGAN, however, aims to avoid such simplifications and create multi-track polyphonic music with” harmonic and rhythmic structure, multi-track interdependency and temporal structure”. Another interesting work in the field of music generation is created by Lijun Zheng and Chenglong Li, Learning Automata-based SA-GAN (LA-SAGAN) [22]. They point out that using the self-attention (SA) version of GAN architecture is quite important for long-distance dependencies in produced pieces. A further advantage of this solution is utilising SA’s mechanism of providing emotional features to the input, which was crucial in the synthesis of music with a specific desired theme, or even genre.

In his work, Korneel van den Broek presented a considerably different solution with his MP3net [23]. It is a WGAN-based deep convolutional model with MP3/Vorbis-like audio compression and Modified Discrete Cosine Transform (MDCT) data representation.

The architecture of the model is a deep 2D convolutional network, where each subsequent generator model block increases resolution along the time axis, also adding higher octave along the frequency axis. Deeper layers are also connected to all the output parts, so the context of the whole music piece is not lost along the way.

2.6. Challenges and limitations

As in any other field of research, there are also many challenges and limitations to overcome when working with music generation and GANs. Those issues can be related to many topics, some of which are not even related to machine learning algorithms themselves.

2.7. Finding dataset

Difficulties may even occur at the very beginning of the process when it comes to data selection. This is a common problem, especially when working on a very specific theme which is not very popular among existing works and datasets [24] [25] [26].

2.8. GANs' common issues

In some use cases, GANs or their modifications may have some limitations due to their distinctive features. It's worth taking a look at some of the most common issues that may occur when working with GANs. Basic RNNs may encounter" vanishing gradient", a problem where weights become too small and result in difficulty in training the network. [27] The generator stops learning because discriminator becomes too effective and labels real and generated data too well, pushing loss function to 0 which results in producing gradients very close to 0 as well [28]. But that's not the only issue with GANs. Korn and Yilmas in their work [29] say that GANs that were introduced in existing works also suffer from issues like:

1. Unstable convergence / Nonconvergence - situation where parameters become unstable, and it's impossible to achieve convergence between generator and discriminator parameters,
2. Mode collapse - generator output is limited to a small subset of samples. It's a common and very challenging GAN problem also known as" the Helvetica scenario" [1]. It happens when the generator learns to trick the discriminator by producing a small variety of outputs. That leads to failure in model training,
3. Overfitting - occurs when the model learns to memorize training data instead of generalizing it. It's a common problem in machine learning and deep learning in general when output is too close to training data [30],
4. Hyperparameters - hyperparameters are not an issue by themselves, but GANs are very sensitive to their selections, which means that even slight changes made to them can result in significant changes in output quality and overall performance.

2.9. Challenges in music generation

Music generation is a very specific field of research, and it has its own challenges that are not present in other fields of research. In Moûsai [26], Schneider et al. propose their diffusion model which should deal with at least some of the challenges they mention in their work including:

- length of the generated music - most of text-to-audio systems were able to generate only short pieces, about few seconds long,
- model efficiency - authors say that many models need to run on several GPUs for hours just to generate one minute of audio,
- lack of output diversity - many models are limited to only single genre of music,
- easy controllability by text prompts - other models are controlled either by latent states, starting snippet of music or text but the lyrics (not the description of piece).

2.10. Evaluation

Another hard topic in the GAN world is their performance and quality evaluation, Korn and Yilmas [29] mention" lack of comprehensive evaluation metric" as one of the biggest problems happening to this type of neural network. Typical measures like model likelihood are not applicable for GANs, moreover there is lack of robust and consistent evaluation metrics and also not many comparisons of models to objectively assess which one has better performance. That's why this is still an active field of research. Trieu and Keller [27] also mention that evaluation of models is a challenge.

2.11. Conclusion

It's visible that peoples' urge to create virtual assistants and generative tools that will boost their daily performance by simplifying certain tasks results in rapid growth in AI field, great development in generative networks and very big interest in every new discovery, every new model which will outperform the others. This results in many new neural network architectures, one of which is the base of this thesis, GAN. GANs popularity and its abundance of usages made an impact on development of many new versions of this neural network like CGAN, DCGAN, WGAN or SAGAN and many more which were trying to improve both the original and GANs' ancestors' architecture. It was also mentioned that music generation using GANs can be achieved in different ways - generating sound waves using sound files, or by composing the music piece in" computer-friendly musical notation" - MIDI, this works main interest. There are many different approaches to music generation, not only the GAN way, but for symbolic music generation it came out to be most effective while maintaining satisfactory performance on consumer-grade devices.

No field of research mentioned related to GANs is fully covered, and there are many possibilities for further studies. That's why it's crucial to compare, evaluate and benchmark existing models - to properly assess their value in possible applications and find the best suiting one for the task, but also to see where further research could be conducted in order to move on forward, make progress and maintain the pace of the remarkable AI growth.

3. Objective

This research's primary objective is investigation of the impact of various GAN models selection on the quality of generated sounds and melodies. It will focus on analysis if GAN can generate melodies that are similar quality to those composed by real musicians and, most importantly, it'll try to answer the question if complexity and architecture of selected models have any impact on the quality of generated pieces and if it does – how significant it is and what's the best approach for generating high quality melodies.

3.1. Thesis

GAN models are capable of generating high-quality musical pieces that are comparable to those composed by humans.

3.2. Hypotheses

For thesis above, the following hypotheses are formulated:

- H1: Can the GAN model generate high quality music that is comparable in quality to human composed melodies?
- H2: Does the complexity of GAN model architecture affect the quality of generated melodies?

4. Tools and research methods

In this research, multiple programs, each using different GAN models, will be used in order to test hypotheses posed in the previous chapter. To ensure minimal impact from external factors, every program will be trained and tested on the same dataset and executed on local PCs with identical hardware specifications.

4.1. Technologies and tools:

4.1.1.Hardware specification:

- CPU – AMD Ryzen 7 7800X3D,
- GPU – NVIDIA GeForce RTX 4070
- RAM memory – 32GB DDR5 6000MHz

Operating system: Ubuntu 24.04.1 LTS

4.1.2.Software specification:

- IDEs:
- IntelliJ IDEA 2025.2
- Visual Studio Code 1.104
- Python 3.10.13 - programming language broadly used in the machine learning world. It is known for its popularity which results in a large number of libraries, also those helpful in creating AI models,
- TensorFlow 2.15.0 - an open-source library for machine learning developed by Google Brain team. Broadly used for neural network and ML applications development.

4.2. Subject and scope of research:

The main subject of this work is the comparison of popular GAN models used for generation of symbolic music based on MIDI files in terms of their quality and performance in preparing synthetic musical pieces that will be as similar to human-composed ones as possible. MIDI was picked as the default music representation file, because it represents music at a symbolic level, making it far closer to how humans conceptualize musical composition. Chosen GAN models are:

- MuseGAN - Created by Dong et al [2]. Goal of this work was to avoid as much of simplifications of previous works as possible in terms of reducing complexity of music pieces by generating only singletrack music, using chronological ordering of notes or generating polyphonic music by combining multiple monophonic tracks,
- MidiNet - created by Yang and his team [31] it is GAN which uses CNN approach rather than RNN. Authors inspired by WaveNet [32], which used the second option and audio files as input rather than symbolic music, wanted to investigate the usage of CNNs for generating melodies. This resulted in model able to create polyphonic pieces with comparable or even better quality than MelodyRNN,
- SeqGAN - introduced by Yu et al. [33] is developed as a response to the challenge that GANs are facing which is generating sequences of discrete tokens. This challenge arose because the discrete outputs of the generator make it very difficult to backpropagate the gradient update from the discriminator. To overcome this problem, the authors proposed their own framework for generating sequential discrete data.

4.3. Dataset:

Lakh MIDI Dataset (LMD) is the dataset chosen for this study. LMD consists of 176 581 MIDI files with music from various genres, from which only 2697 rock pieces were filtered out, which is very similar approach to MuseGAN paper, where authors used Lakh Pianoroll Dataset which is LMD but in the form of pianorolls saved as numpy arrays.

4.4. Evaluation metrics:

Experimental parameters used in this research have to clearly signal which GAN architecture is best for the quality of generated melodies.

Mode collapse metrics - quality of generated music in general:

- Qualified Rhythm frequency (QR) - measure of how often lengths of notes are within valid beat ratios, which are {1, 1/2, 1/4, 1/8, 1/16}. Also, their dotted and triplet counterparts are included, and any combination of two of those ratios,
- Consecutive Pitch Repetitions (CPR) - measure of how often the same pitch is repeated l times in a row, where l is a specified number of repetitions that should be taken into account. We've chosen l = 8 because in rock music there could be more repetitions because of simplicity of some of the rock subgenres,
- Durations of Pitch Repetitions (DPR) - measure of how frequently the same pitch is repeated for a certain minimum duration of time. In this case, the threshold value was 24 timesteps, which is equivalent to 2 melody bars.
- Off-beat Recovery frequency (OR) – measure of how often a generated note, after a time, offset of d, falls back on beat-aligned positions (e.g., every 6 steps for eighth notes in 48-timestep bars).

Creativity Metrics:

- Tone Spans (TS) – how many neighbouring notes had tonal difference of single octave,
- Pitch Variations (PV) - measure of how many different pitches are used in generated sequence,

- Rhythm Variations (RV) - measure of how many different durations of a note are played within a generated sequence.

5. Results

Each metric was calculated for a single track from sample data and then average value for samples for training dataset, and both models were calculated as a final result for each.

Unfortunately, results for SeqGAN are not present in the table, because of problems with configuration of the model. Among many trials there wasn't a single one in which model would remain stable and present healthy competition between generator and discriminator neural networks.

Table 1: Results of tests conducted for this article

METRIC	Dataset	MidiNet	Muse-GAN
QR	0.45521	0.02162	0.22442
CPR	0.02873	0.99611	0.00685
DPR	517.25139	128.00000	3.42188
OR	0.12447	0.12500	0.12099
TS	0.00038	0.00002	0.00897
PV	0.00472	0.03113	0.40853
RV	0.00155	0.00049	0.07210

6. Discussion

The dataset had ~46% of notes with lengths that were valid for beat ratios from our list. MidiNet performed very poorly with just 2% of such, meaning MuseGAN's definite advantage with a result of 22%.

CPR for training dataset was about 0.02873, meaning that ~3% of the data included repetitions of single pitch at least 8 times in a row. MuseGAN performed definitely better than MidiNet, values of this metric for models were 0.99611 and 0.00685 accordingly meaning that the second one's structure of generated track was more alike with training data. The value for MidiNet suggests mode collapse.

Input data's DPR was ~517 which means high occurrence of repetitions of long-lasting notes. MidiNet value was somewhat near with a result of 128. MuseGAN on the other hand had a 3.42188 DPR value which means a more varied nature of samples generated by this model.

OR for every set of data was nearly the same with 0.12447 for training data, 0.12500 for MidiNet and 0.12099 for MuseGAN meaning low Off-beat Recovery frequency for each.

In terms of creativity metrics, MuseGAN performed the best with the highest value for each. This model's TS value was about 0.00897 meaning most frequent appearance of at least octave difference between two consecutive notes, significantly higher than input data and MidiNet, for which those values were rather small – 0.00038 and 0.00002 accordingly.

MuseGAN has shown also the most varied pitches among compared models wth PV value of 0.40853, significantly higher than MidiNet and LMD with 0.03113 and 0.00472 accordingly.

The same situation happened with the Rhythm Variations metric. MuseGAN generated the most differentiated samples in terms of note lengths.

Overall, the second model outperformed MidiNet in terms of numerical results. The greater diversity of generated pieces and significantly higher resistance to mode collapse were decisive factors that indicated its superiority.

7. Conclusions

SeqGAN evaluation was not possible due to problems with overall stability. The hyperparameters chosen in every test suite didn't allow for even a single positive learning run which could be the fault of values chosen for each.

Both models that were possible to compare performed very differently for the dataset of choice, which confirms the second hypothesis that complexity of model architecture affects the quality of generated melodies.

MuseGAN has shown higher creativity in generated tracks and was more resistant to mode collapse. MidiNet on the other hand, showed a high probability of occurrence of such issues which was mostly visible in the CPR metric. Overall MidiNet seemed to generate closer samples to input data but most probably due to low creativity and mode collapse occurrence. That leads to the conclusion that MuseGAN is a better model overall.

Results for this model show that GAN models can produce differentiated and original melodies, however that doesn't mean they're human-grade quality, because of complexity of the musical pieces and their characteristic parts among single song of certain genre, which couldn't be reflected by evaluated models because of the way they've processed each track, what counterparts both thesis and first hypothesis of this work.Future directions

Future work could include a broader selection of models, including those based on other architectures such as diffusion models. Featuring more models may be even more impactful for the overall discussion about music generation topic. These types of papers are invaluable during research and development processes while working on new or improved solutions.

Evaluation of models could, just like some of the works presented in the literature review section, also incorporate user study, that may allow for an evaluation of the models from the human perspective, be it casual listeners of even professional musicians. These types of studies, featuring humans listening to created music, are crucial for a real and grounded improvement in this subject as, in the end, music is supposed to be listened to and enjoyed by humans.

We hope that this paper, as a thorough comparison of some of the most popular models, can serve as a foundation for further expansion and advancement of presented topics.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, Advances in Neural Information Processing Systems 27 (2014) 2672-2680, https://doi.org/10.48550/arXiv.1406.2661.

[2] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, Y.-H. Yang, MuseGAN: Multitrack sequential generative adversarial networks for symbolic music generation and accompaniment, arXiv preprint arXiv:1709.06298 (2017), https://doi.org/10.48550/arXiv.1709.06298.

[3] M. Mirza, S. Osindero, Conditional Generative Adversarial Nets, arXiv preprint arXiv:1411.1784 (2014), https://doi.org/10.48550/arXiv.1411.1784.

[4] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv preprint arXiv:1511.06434 (2016), https://doi.org/10.48550/arXiv.1511.06434.

[5] E. Denton, S. Chintala, A. Szlam, R. Fergus, Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks, Advances in Neural Information Processing Systems 28 (2015) 1486–1494, https://doi.org/10.48550/arXiv.1506.05751.

[6] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, arXiv preprint arXiv:1703.10593 (2017), https://doi.org/10.48550/arXiv.1703.10593.

[7] H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-Attention Generative Adversarial Networks, Proceedings of the 36th International Conference on Machine Learning (2019) 7354–7363, https://doi.org/10.48550/arXiv.1805.08318.

[8] N. Kodali, J. Hays, J. Abernethy, Z. Kira, On Convergence and Stability of GANs, arXiv preprint arXiv:1705.07215 (2017), https://doi.org/10.48550/arXiv.1705.07215.

[9] L. Barillaro, Deep Learning Platforms: Keras, PyTorch, Tensorflow, Reference Module in Life Sciences, Elsevier, 2024, B9780323955027001676, B9780323955027000932, B9780323955027000920, https://doi.org/10.1016/B978-0-323-95502-7.00092-0, https://doi.org/10.1016/B978-0-323-95502-7.00093-2, https://doi.org/10.1016/B978-0-323-95502-7.00167-6.

[10] M. Saito, E. Matsumoto, S. Saito, Temporal Generative Adversarial Nets with Singular Value Clipping, arXiv preprint arXiv:1611.06624 (2016), https://doi.org/10.48550/arXiv.1611.06624.

[11] A. Schnepf, F. Vasile, U. Tanielian, 3DGEN: A GAN-Based Approach for Generating Novel 3D Models from Image Data, arXiv preprint arXiv:2312.08094 (2023), https://doi.org/10.48550/arXiv.2312.08094.

[12] C. Vondrick, H. Pirsiavash, A. Torralba, Generating Videos with Scene Dynamics, arXiv preprint arXiv:1609.02612 (2016), https://doi.org/10.48550/arXiv.1609.02612.

[13] H. Gao, X. Yang, Y. Hu, B. Wang, H. Xu, Z. Liang, H. Mu, Y. Wang, Y. Chen, GANs-generated Synthetic Datasets for Face Alignment Algorithms in Complex Environments, Applied Soft Computing 151 (2024) 112260, https://doi.org/10.1016/j.asoc.2024.112260.

[14] D. Efimov, D. Xu, L. Kong, A. Nefedov, A. Anandakrishnan, Using Generative Adversarial Networks to Synthesize Artificial Financial Datasets, arXiv preprint arXiv:2002.02271 (2020), https://doi.org/10.48550/arXiv.2002.02271.

[15] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T. S. Huang, Generative Image Inpainting with Contextual Attention, arXiv preprint arXiv:1801.07892 (2018), https://doi.org/10.48550/arXiv.1801.07892.

[16] K. Schawinski, C. Zhang, H. Zhang, L. Fowler, G. K. Santhanam, Generative Adversarial Networks Recover Features in Astrophysical Images of Galaxies Beyond the Deconvolution Limit, Monthly Notices of the Royal Astronomical Society Letters 467(1) (2017) L110–L114, https://doi.org/10.1093/mnrasl/slx008.

[17] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, X. Tang, ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks, arXiv preprint arXiv:1809.00219 (2018), https://doi.org/10.48550/arXiv.1809.00219.

[18] R. V. Bisneto, A. O. Carvalho Filho, D. M. V. Magalhães, Generative Adversarial Network and Texture Features Applied to Automatic Glaucoma Detection, Applied Soft Computing 91 (2020) 106165, https://doi.org/10.1016/j.asoc.2020.106165.

[19] Y. Mirsky, T. Mahler, I. Shelef, Y. Elovici, CT-GAN: Malicious Tampering of 3D Medical Imagery Using Deep Learning, arXiv preprint arXiv:1901.03597 (2019), https://doi.org/10.48550/arXiv.1901.03597.

[20] J. Rothstein, MIDI: A Comprehensive Introduction, A-R Editions, Madison, 1995.

[21] Multimedia Programming Interface and Data Specifications 1.0, TEX–basics, https://www.tactilemedia.com/info/MCI_Control_Info.html, [12.01.2025].

[22] L. Zheng, C. Li, Real-Time Emotion-Based Piano Music Generation Using Generative Adversarial Network (GAN), Proceedings of the IEEE (2024), https://ieeexplore.ieee.org/document/10557596.

[23] K. van den Broek, MP3net: Coherent, Minute-Long Music Generation from Raw Audio with a Simple Convolutional GAN, arXiv preprint arXiv:2101.04785 (2021), https://doi.org/10.48550/arXiv.2101.04785.

[24] M. Zheng, P. Bai, X. Shi, X. Zhou, Y. Yan, FT-GAN: Fine-Grained Tune Modeling for Chinese Opera Synthesis, Proceedings of the AAAI Conference on Artificial Intelligence 38(17) (2024) 29943, https://doi.org/10.1609/aaai.v38i17.29943.

[25] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, Y.-H. Yang, EMOPIA: A Multi-Modal Pop Piano Dataset for Emotion Recognition and Emotion-Based Music Generation, arXiv preprint arXiv:2108.01374 (2021), https://doi.org/10.48550/arXiv.2108.01374.

[26] F. Schneider, O. Kamal, Z. Jin, B. Schölkopf, Moûsai: Text-to-Music Generation with Long-Context Latent Diffusion, arXiv preprint arXiv:2301.11757 (2023), https://doi.org/10.48550/arXiv.2301.11757.

[27] N. Trieu, R. Keller, JazzGAN: Improvising with Generative Adversarial Networks, Proceedings of the 21st International Conference on Principles and Practice of Constraint Programming (2018), https://www.semanticscholar.org/paper/JazzGAN-%3A-Improvising-with-Generative-Adversarial-Trieu-Keller/1ccaf621cf9a9d25ee4f81dd287af3b0fd2c7989.

[28] M. Wiatrak, S. V. Albrecht, Stabilizing Generative Adversarial Network Training: A Survey, arXiv preprint arXiv:1910.00927 (2019), https://doi.org/10.48550/arXiv.1910.00927.

[29] B. Yilmaz, R. Korn, A Comprehensive Guide to Generative Adversarial Networks (GANs) and Application to Individual Electricity Demand, Expert Systems with Applications 247 (2024) 123851, https://doi.org/10.1016/j.eswa.2024.123851.

[30] Y. Yazici, C.-S. Foo, S. Winkler, K.-H. Yap, V. Chandrasekhar, Empirical Analysis of Overfitting and Mode Drop in GAN Training, arXiv preprint arXiv:2006.14265 (2020), https://doi.org/10.48550/arXiv.2006.14265.

[31] L.-C. Yang, S.-Y. Chou, Y.-H. Yang, MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation, arXiv preprint arXiv:1703.10847 (2017), https://doi.org/10.48550/arXiv.1703.10847.

[32] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A Generative Model for Raw Audio, arXiv preprint arXiv:1609.03499 (2016), https://doi.org/10.48550/arXiv.1609.03499.

[33] L. Yu, W. Zhang, J. Wang, Y. Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, Proceedings of the AAAI Conference on Artificial Intelligence 31(1) (2017) 2852-2858, https://doi.org/10.48550/arXiv.1609.05473.