JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 39 (2026) 115–122
Received: 29 October 2025
Accepted: 9 February 2026

Generative Adversarial Networks in Sound Synthesis: Analysis of Sound Modeling Capabilities Using GANs

Michał Galant*, Paweł Powroźnik

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

Abstract

This paper explores the capabilities of Generative Adversarial Networks (GANs) for audio synthesis. Particularly, it focus on WaveGAN, which works directly with raw waveforms, and DCGAN trained on spectrogram representations. These models were trained using internal combustion engine sounds collected from the Google AudioSet corpus. The study aims to assess how data representation and architectural choices affect not only quality but also training dynamics in synthesis. It exposed different behavioral characteristics of both approaches thus underlining the fact that signal representation largely defines the generative process and perceptual properties of output. Modelling on spectrograms, even though phase information is naturally lost, proved more robust and stable than waveform generation directly from waveforms. The results have further affirmed that a fair share of decisive synthesis quality with GAN-based audio synthesis resides in factors including data complexity, the domain of representation, and hyperparameter configuration.

Keywords: GAN; sound synthesis; audio generation; WaveGAN; DCGAN

*Corresponding author

Email address: **s95401@pollub.edu.pl** (M. Galant)

1. Introduction

The rapid advancement of artificial intelligence has highlighted the prominence of Generative Adversarial Networks (GANs) [1]. Unlike Large Language Models, GANs focus primarily on the autonomous generation of novel data, integrating content synthesis with advanced pattern recognition [2]. This study empirically evaluates the effectiveness of GANs for audio synthesis, specifically examining how different input data representations impact the quality of the generated signal.

To address the challenge of modelling complex acoustic structures, two hypotheses were formulated:

H1: Generative Adversarial Networks are an effective tool for audio synthesis tasks.

H2: The input data representation significantly impacts the quality of the generated audio samples.

The paper compares two distinct architectures: WaveGAN (raw waveforms) and DCGAN (spectrograms), using the AudioSet dataset [3]. The main contribution lies in the comparative analysis of these representations under challenging training conditions, highlighting that spectrogram-based approaches can offer greater resilience to catastrophic model failure than direct waveform synthesis.

2. Literature review

Although Generative Adversarial Networks (GANs) have dominated discussions on content synthesis, it is essential to first consider the contributions and capabilities of other neural architectures in this domain.

2.1. Neural networks in sound synthesis

Recurrent Neural Networks (RNNs) have been effectively applied to audio generation, as demonstrated by the RNN-WaveNet model proposed by Kim et al. [4]. This approach integrates RNNs with a WaveNet vocoder, allowing fine-grained control over timbre through instrument embeddings and smooth interpolation within the latent space.

Bazin et al. [5] introduced a VQ-VAE-2 architecture for interactive musical instrument synthesis, employing spectrogram inpainting and transformer-based token masking to reconstruct missing spectral regions. Their system demonstrated real-time control capabilities and high efficiency on the NSynth dataset.

Deep autoencoders have also been utilized to emulate nonlinear audio effects, effectively capturing temporal dependencies and analog-like nonlinearities [6]. Although these models enable the creation of novel effects from curated datasets, issues such as noisy outputs and generalization remain open challenges.

2.2. Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [1], have become a cornerstone of generative machine learning. A GAN consists of a generator that synthesizes data and a discriminator that evaluates authenticity, forming a minimax optimization process that drives mutual improvement and enables unsupervised learning of complex data distributions.

Despite their success, GANs are difficult to train due to instability and mode collapse, where the generator produces repetitive or low-diversity samples. To mitigate these issues, variants such as Wasserstein GAN (WGAN) and WGAN-GP, as well as stabilization techniques like spectral normalization, label smoothing, and controlled noise injection, have been proposed [2, 7].

GANs have been applied across multiple modalities, including image, speech, and music generation, as well as neural decoding from brain signals [8]. In audio synthesis, convolutional and conditional GANs are particularly relevant, as they can model intricate temporal and spectral patterns. Common evaluation metrics include the

Inception Score (IS), Fréchet Inception Distance (FID), and Wasserstein Distance, complemented by expert perceptual assessments [2, 7].

2.3. GANs in sound synthesis

The application of Generative Adversarial Networks (GANs) to audio synthesis has rapidly advanced, encompassing both waveform and spectrogram domains. Early architectures such as WaveGAN and SpecGAN [9] demonstrated the feasibility of adversarial learning for raw audio, though they faced limitations due to high sampling rates and computational cost.

Later developments, including MelGAN [10], GAN-Synth [11], HiFi-WaveGAN [12], FGP-GAN [13], and EVA-GAN [14], introduced convolutional generators, multi-scale discriminators, and spectral or phase-based losses, enabling real-time synthesis and enhanced perceptual quality. Specialized models have been applied to speech, singing, and instrumental sound synthesis, with WGANSing [14], DrumGAN [15], and EnvGAN [16] representing notable examples.

Training stability and perceptual realism were further improved through multi-resolution discrimination, spectral normalization, and loss functions tailored to human auditory perception. Attention-based mechanisms have proven effective in capturing long-term dependencies [17], while modern systems integrate GAN models into production workflows as VST plugins or modular environments [18].

Recent research trends emphasize hybrid and multimodal architectures, self-supervised learning, and interactive synthesis tools allowing real-time control [19, 15]. From early designs like WaveGAN to contemporary conditional models such as MelGAN and DrumGAN, GANs continue to evolve into powerful and versatile instruments for sound generation.

3. Materials and methods

To test the formulated hypotheses, a comparative experiment was conducted. A dataset of vehicle engine sounds was curated from the Google AudioSet collection, undergoing extensive preparation that included slicing, augmentation, and filtering. Subsequently, two distinct GAN architectures were implemented and trained on this dataset: WaveGAN, which operates directly on raw audio waveforms, and DCGAN, which learns from 2D spectrogram images.

The effectiveness of both models was assessed through a combination of objective quality metrics and subjective perceptual analysis of the generated audio. The experiment was conducted on a high-performance workstation to ensure efficient, multi-epoch training. The key hardware component was an NVIDIA GeForce RTX 4090 GPU, supported by an AMD Ryzen Threadripper PRO 7965WX CPU and 512 GB of DDR5 RAM. The implementation was developed in Python 3.11.9 using the PyTorch framework. Core libraries for the experiment included torchaudio for audio signal processing and model implementation, alongside numpy, pandas, and scipy for data manipulation and analysis.

3.1. Dataset preparation

The experiment utilized the Google AudioSet, a collection of over 2 million 10-second audio samples from YouTube, to create a specialized subset for synthesizing vehicle engine sounds. The selection resulted with a total of 14 classes related to engine operations, such as *engine, medium_engine_mid_frequency, vehicle, truck, engine_knocking, engine_starting, motorcycle, motor_vehicle_road, car, rumble, heavy_engine_low_frequency, light_engine_high_frequency, accelerating_revving_vroom* and *race_car_auto_racing*, giving in an initial set of 1,881 samples.

This dataset underwent a rigorous six-stage processing pipeline to enhance its quality and diversity for training the GAN models:

1. **Source Data Selection and Filtering**: Recordings from the selected classes were extracted, and irrelevant samples were removed.
2. **Sample Slicing**: The 10-second clips were divided into 2-second segments with a 1-second overlap to increase the number of examples and stabilize the learning process, resulting in 16,780 segments.
3. **Data Augmentation**: To improve model generalization, controlled random transformations were applied, including pitch shifting, volume adjustment, and spectrogram masking. This expanded the dataset to over 167,000 examples.
4. **Signal Filtering**: Samples with excessively low or high volume, high noise levels, or atypical frequency characteristics were removed based on metrics like RMS and Spectral Flatness. This rigorous filtering rejected approximately 51.1% of the augmented data, reducing the dataset to 82,109 samples.
5. **Voice Activity Detection (VAD) Filtering**: The Silero VAD model was used to eliminate samples containing human speech, ensuring the domain purity of the dataset. This step removed an additional 11,586 samples.
6. **Spectrogram Generation and Finalization**: For each audio file, a corresponding spectrogram and metadata file were prepared.

After this comprehensive preparation, the final dataset consisted of 70,523 audio samples, providing the necessary acoustic consistency and diversity to effectively train the WaveGAN and DCGAN architectures.

3.2. WaveGAN and DCGAN implementation

The WaveGAN architecture was implemented to analyze the capabilities of generative models that operate directly on raw audio signals. Unlike spectrogram-based approaches, WaveGAN generates 1D time-domain waveforms, which avoids the information loss that results from transforming the signal into a two-dimensional representation. The model was based on the Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) framework, which enables more stable training and mitigates the phenomenon of mode collapse.

The implementation involves two competing networks:

- a **generator**, which transforms a random noise vector into an audio sample,
- a **discriminator**, which assesses the authenticity of a sample through binary classification.

The generator was constructed using 1D transposed convolutional layers with ReLU activations, which progressively upsample the signal to the target length (44,100 values, corresponding to 2 seconds of audio at a 22.05 kHz sampling rate). The discriminator has a mirrored structure, employing standard 1D convolutional layers and LeakyReLU activation functions. A key feature of the implementation is its modular configuration via JSON files, which allows for flexible adjustment of model parameters. The configuration includes, among other things:

- paths to the dataset and output directories,
- the size of the latent vector and the length of the samples,
- the number of epochs, batch size, and learning rate,
- the frequency of saving checkpoints and generating control samples,
- parameters of the loss function, including the weight of the gradient penalty.

The DCGAN (Deep Convolutional Generative Adversarial Network) architecture was implemented as the second of the analyzed structures, in order to compare the effectiveness of generative models that operate on visual representations of sound. In contrast to WaveGAN, DCGAN processes spectrograms-two-dimensional images that visualize the signal's energy in the time-frequency domain. The generator's structure comprises a sequence of 2D transposed convolutional layers that successively increase the image resolution. The discriminator is built from standard 2D convolutional layers with LeakyReLU activations and layer normalization. The training process was based on the Binary Cross-Entropy (BCE) loss function.

The model implementation allows for a wide range of configurations, defined in external JSON files. The parameters include:

- the size of the latent vector (typically 100-256),
- the dimensions of the input spectrograms,
- the number of convolutional layers and filters,
- the learning rates for the generator and discriminator,
- the frequency of saving results and checkpoints,
- paths to the data and results directories.

Thanks to its modular structure and the ability to dynamically adjust hyperparameters, DCGAN served as a contrasting reference point to WaveGAN, allowing for an evaluation of how data representation affects training stability and the perceptual quality of the generated samples. A summary of the key differences between the models is presented in Table 1.

Table 1: Comparison of key aspects in the implementation of architectures

Aspect	DCGAN	WaveGAN
Data type	2D Spectrograms (128×128 px)	1D Audio (44100 samples)
Convolutions	Conv2d/ConvTranspose2d	Conv1d/ConvTranspose1d
Kernel size	4×4	25
Loss function	Binary Cross-Entropy	Wasserstein + Gradient Penalty
Training strategy	1:1 (G:D)	n_critic=5 (1:5)
Optimizer	Adam (lr=0.0001)	Adam (lr=0.0001)
Latent dim	128-256	100-128
Batch size	384	64
Number of epochs	800	8000
Preprocessing	Image normalization	Audio resampling
Output	Spectrogram images	Raw audio samples

3.3. Measures of effectiveness and efficiency for the implemented architectures

The evaluation of the WaveGAN and DCGAN models' performance required the application of diverse measures, encompassing both quantitative (numerical) and qualitative (perceptual) aspects. The objective of these measurements was to determine the extent to which the audio samples generated by the networks corresponded to the characteristics of real recordings, as well as to assess the stability and efficiency of the training process itself.

To quantitatively evaluate model performance, a dedicated analysis module was developed to compute and log key metrics throughout the training process. The selected metrics were divided into two categories: universal indicators for monitoring GAN training dynamics and data-specific measures for assessing the quality of the generated samples.

Universal training metrics included the generator and discriminator loss functions, used to track the adversarial learning process, and gradient norms, which served as a crucial indicator of training stability. Monitoring gradient norms was essential for diagnosing common training problems such as vanishing gradients (values approaching zero) or exploding gradients (rapidly increasing values).

For the evaluation of generated samples, distinct metrics were applied based on the data representation:

- for raw audio signals (WaveGAN), the primary metrics were the Fréchet Audio Distance (**FAD**), which measures the perceptual quality by comparing feature distributions of real and synthetic audio, and the Signal-to-Noise Ratio (**SNR**), which quantifies the level of noise relative to the useful signal. These were combined into a composite Audio Quality Score (**AQS**), calculated as a weighted linear combination:

$$AQS = 0.6 \cdot FAD_{norm} + 0.4 \cdot SNR_{norm} \qquad (1)$$

To ensure the score reflects quality on a 0-100 scale, the FAD metric (where lower is better) was inverted using the formula $FAD_{norm} = 100 - FAD_{score}$, while the SNR (where higher is better) was normalized by scaling the decibel value: $SNR_{norm} = \frac{SNR_{dB}}{60} \cdot 100$.

- for spectrogram images (DCGAN), evaluation was based on the Fréchet Inception Distance (**FID**) and the Inception Score (**IS**), which measures both the quality and diversity of the output images. Similarly, these were aggregated into a composite Image Quality Score, prioritizing the distance metric with a 70% weight:

$$IQS = 0.7 \cdot FID_{norm} + 0.3 \cdot IS_{norm} \quad (2)$$

The normalization process involved scaling down the FID metric ($FID_{norm} = \max(0, 100 - \frac{FID}{3})$) and linearly scaling the Inception Score ($IS_{norm} = \max(0, (IS - 1) \cdot 10)$) to create a unified indicator of visual generation performance.

These applied measures provided the data for an objective comparison of the architectures. The results showed that WaveGAN exhibited greater computational requirements but generated signals with a richer temporal structure. In contrast, DCGAN provided higher training stability at a lower computational cost, at the expense of losing some phase information. Thus, the evaluation made it possible not only to assess the quality of the results but also to analyze the relationship between data representation, training stability, and the operational efficiency of each model.

3.4. Monitoring the training process

Effective monitoring was essential for ensuring experimental stability and enabling detailed post-hoc analysis. A comprehensive logging system tracked quantitative metrics-including generator and discriminator loss functions, iteration counts, and epoch durations-supplemented by the periodic generation of control audio samples for qualitative assessment.

The system also monitored hardware resource utilization (CPU, GPU, RAM) to detect potential overloads. A critical component was the automated checkpointing mechanism, which regularly saved model states and hyperparameters. This feature ensured reproducibility and allowed for resuming the training process or analyzing results from any specific stage of the experiment.

4. Results and discussion

The objective of the analysis was to provide an objective presentation of the training results for the WaveGAN and DCGAN models and to verify the hypotheses regarding their effectiveness in audio synthesis. The analysis comprised three stages: an evaluation of the training process stability, an analysis of quantitative results, and a perceptual evaluation of the generated samples.

The first part focuses on the course of the learning process and the stability of the architectures, utilizing the saved logs of loss function values and gradient norms, which are primary indicators of training progress. The subsequent sections present the results of qualitative metrics and perceptual comparisons of the generated samples with the source data.

4.1. Course and Stability of the Learning Process

The stability and dynamics of the learning process are a crucial aspect of the effectiveness of GAN models, whose operation is based on the competition between a generator and a discriminator. The analysis of the logs revealed that both models encountered significant convergence problems, although these were of a different nature.

For the WaveGAN model, the loss function values for the generator and discriminator (Figure 1) exhibited initial stability for the first 14 epochs. This was followed by a sharp increase in loss to the order of 10^7, indicating the occurrence of a loss explosion. The implemented monitoring mechanism, with a *loss_explosion_threshold* of 200.0, correctly detected the anomaly and interrupted further optimization steps, logging the warning: "Training stopped due to convergence: diverged."

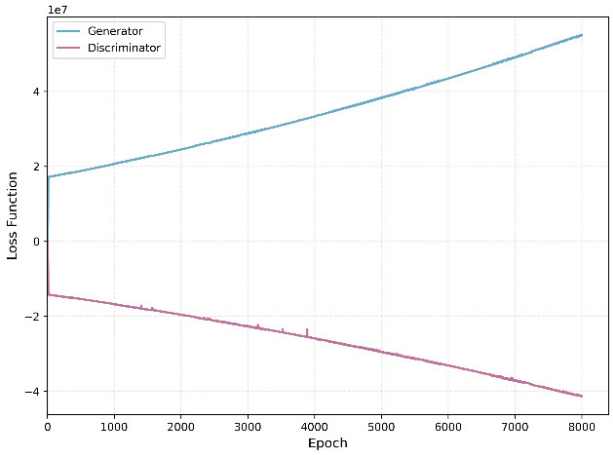

Figure 1: Generator and discriminator loss functions for the WaveGAN model

Simultaneously, an analysis of the gradient norms revealed that their values were 0.0 throughout the entire process, confirming the occurrence of gradient vanishing. This indicates a complete absence of a learning signal-despite large errors, the model was unable to update its weights, and the learning process was effectively halted.

An analysis of the epoch duration confirmed the consequences of this collapse. Initially, a single epoch took approximately 83 minutes, whereas after the 15th epoch, this time was significantly reduced due to the automatic interruption of the process upon error detection. Nevertheless, completing 8000 epochs took over 90 hours.

In the case of DCGAN, the loss function values (Figure 2) remained within a stable range, which initially suggested that the training was proceeding correctly. However, an analysis of the gradient norms revealed that in this case as well, the gradients were 0.0 for the entire duration of the experiment. This means that despite the apparent stability of the loss function, the model was not learning, and the observed oscillations were the result of

random fluctuations rather than actual weight changes. This phenomenon was termed hidden stagnation.

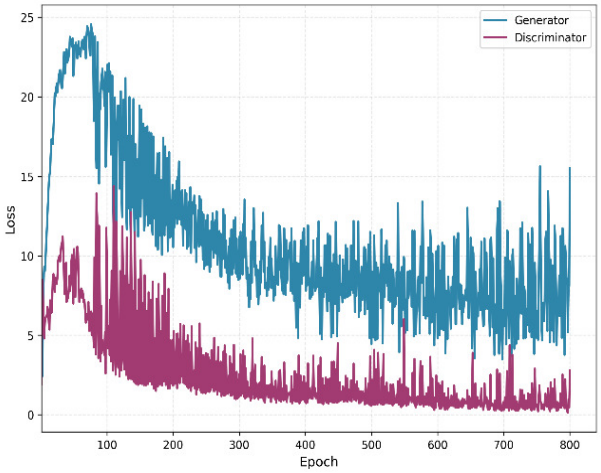

Figure 2: Generator and discriminator loss functions for the DCGAN model

A summary of the most important parameters and characteristics of the learning process is presented in Table 2. The WaveGAN model required ~90 hours of training for 8000 epochs, whereas the DCGAN completed 800 epochs in approximately 40 minutes. These differences arise from different hyperparameters and the characteristics of the input data-raw audio signals require significantly more computational power than spectrogram images.

Table 2: Hyperparameters from the configuration of both models

Parameter	WaveGAN value	DCGAN value
Number of epochs	8000	800
Training time	~ 90 hours	~ 40 minutes
latent_dim	128	128
batch_size	96	384

4.2. Quantitative results: evaluation using effectiveness metrics

Following the analysis of the problematic training process of both models, this section presents the quantitative results describing the quality of the samples generated during the experiment. Although neither model underwent successful training, the collected values provide information about the state of the models at the moment of collapse (for WaveGAN) or stagnation (for DCGAN). The original sample dataset was adopted as a baseline for the audio metrics, for which the average values are presented in Table 3.

Table 3: Key metric values for the original sample dataset

Metric	Value
FAD – average	14.6
FAD – standard deviation	1.64
SNR – average	9.23 dB
SNR – standard deviation	2.47 dB
Audio Quality Score – average	57.4
Audio Quality Score – standard deviation	0.79

The WaveGAN model, which collapsed at the 15th epoch, generated samples of very low quality. The final metric values, reflecting the state of the model just before the loss of stability, are collected in Table 4.

Table 4: Key metric values after WaveGAN training

Metric	Value
FAD	999
SNR	32.04 dB
Audio Quality Score	21.36

The evolution of the individual indicators during the unstable training shows that the model exhibited no tendency for improvement. The Fréchet Audio Distance (FAD) value remained at a constant value of 999, which confirms a large statistical discrepancy between the generated and real samples. The Signal-to-Noise Ratio (SNR) consistently assumed negative values, indicating the dominance of noise (Figure 3). As a result, the composite Audio Quality Score showed no upward trend.

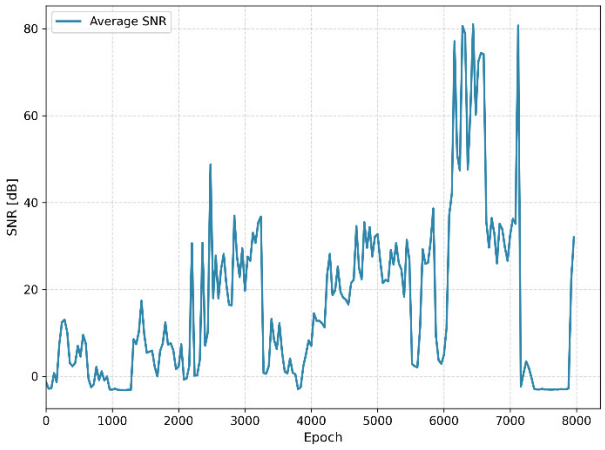

Figure 3: SNR metric evolution over the training epochs for WaveGAN

The DCGAN model, whose training was characterized by stagnation, generated spectrogram images for 800 epochs. Their quality, evaluated using visual metrics, is summarized in Table 5.

Table 5: Final quality metrics for spectrograms generated by DCGAN

Metric	Value
FID	999
Inception Score	4.84
Image Quality Score	11.53
FAD	0.21
SNR	-2.98 dB
Audio Quality Score	59.87

The lack of learning progress is reflected in the evolution of the visual metrics. The Fréchet Inception Distance (FID) value remained at a constant, high value of 999 throughout the training, indicating no improvement in the realism of the generated images. The Inception Score (IS) oscillated within a narrow range, indicating the generation of samples with low and unchanging quality and diversity (Figure 4).

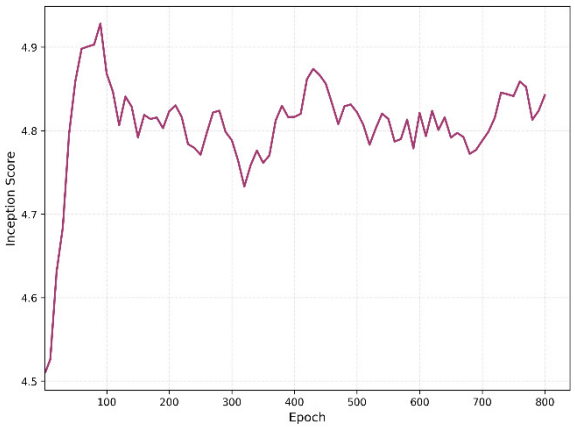

Figure 4: IS metric evolution over the training epochs for DCGAN

Consequently, the Image Quality Score remained at a low level of approximately 11 points.

After converting the generated spectrograms into audio files, they were evaluated using audio metrics. Despite the training stagnation, the values of these metrics also oscillated at a relatively constant level.

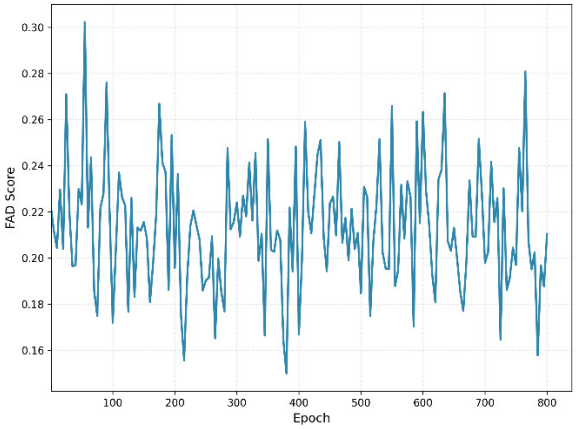

Figure 5: FAD metric evolution over the training epochs for DCGAN (after spectrogram-to-audio conversion)

The FAD value for the reconstructed audio remained in the 0.16- 0.3 range (Figure 5), the SNR oscillated around - 3 dB (Figure 6), and the Audio Quality Score stayed at a level of approximately 59 points.

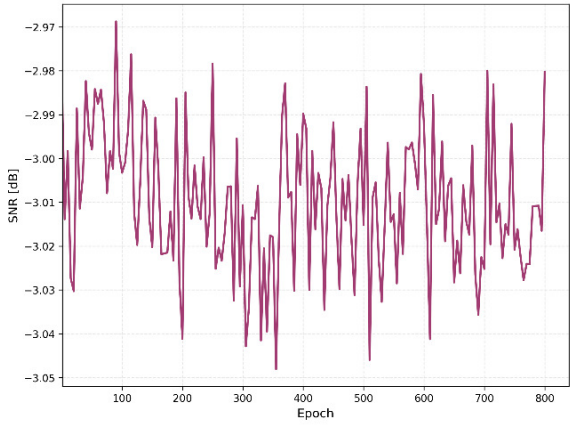

Figure 6: SNR metric evolution over the training epochs for DCGAN (after spectrogram-to-audio conversion)

A key element of the experiment is the direct comparison of the audio signal quality obtained from both architectures. The results of this comparison, along with the baseline values from the original dataset, are compiled in Table 6.

Table 6: Metrics between the models and the original dataset

Metric	Dataset value	WaveGAN value	DCGAN value
FAD	14.6	999	0.21
SNR	9.23 dB	32.04 dB	-2.98 dB
Audio Quality Score	57.4	21.36	59.87

These data provide interesting conclusions. Although both models failed, their failure modes translated into drastically different quality outcomes. The WaveGAN model, which underwent a sudden collapse, generated samples with an extremely high FAD value and a very low Audio Quality Score. Surprisingly, the DCGAN model, despite its complete stagnation, generated images whose reconstructed audio form achieved significantly better results. The FAD value is several orders of magnitude better than that of WaveGAN, and the Audio Quality Score is incomparably higher. This suggests that even an untrained DCGAN generator produced noise with a structure more akin to spectrograms than the WaveGAN generator's noise was to raw audio waveforms.

4.3. Qualitative results: perceptual analysis

Beyond the quantitative evaluation, a key element of the analysis is the qualitative assessment of the generated samples, which includes a visual analysis of spectrograms and a subjective, perceptual evaluation of the sound. This stage allows for an understanding of the nature of artifacts and imperfections that cannot always be fully captured by metrics.

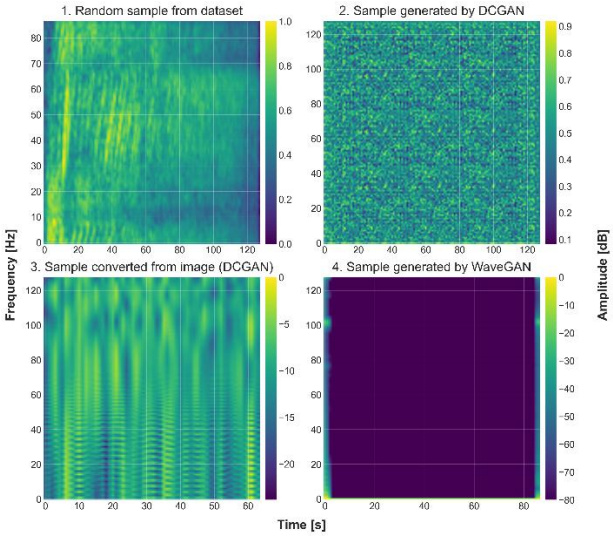

Figure 7: Visual spectrograms: original sample, sample generated and reconstructed by DCGAN, and sample generated by WaveGAN

A visual comparison of the spectrograms (Figure 7) immediately highlights the fundamental differences in quality between the original data and the samples

generated by both models. The original spectrograms are characterized by a complex structure with clearly visible harmonic bands and dynamic changes over time. In contrast, the spectrograms generated by DCGAN are almost entirely devoid of detail, resembling a replicated, simple visual pattern dominated by blurred bands and blocky artifacts. The spectrograms obtained from the WaveGAN samples also exhibit significant degradation, presenting an image of chaotic, broadband noise.

The listening impressions confirm the conclusions from the visual and quantitative analyses. Neither model generated audio that resembled the training dataset; however, the nature of the failure in each architecture was diametrically different. Listening to samples from the initial epochs of WaveGAN training revealed primarily chaotic noise enriched with clicking artifacts. In the middle phase of the training, a certain repetitive, rhythmic audio artifact could be discerned, but after the process collapsed (from approximately the 5000th epoch), the generated samples were nothing but absolute silence. The audio reconstructed from the DCGAN spectrograms had a completely different character-it can be described as a uniform, metallic humming sound, devoid of dynamics and low frequencies. Artifacts typical of the audio reconstruction process, such as phasing effects and synthetic reverberation, were also observed.

The qualitative observations also made it possible to assess the occurrence of mode collapse. In the case of the WaveGAN model, classic mode collapse was not observed; the samples, although of poor quality, showed some diversity, and the final state was silence, which is a form of catastrophic failure. The DCGAN model unequivocally succumbed to this phenomenon. Both the visual and auditory analyses confirmed that throughout the entire training period, the model generated practically identical, indistinguishable samples. All spectrograms displayed the same pattern, and the audio reconstructed from them had an identical, metallic timbre.

5. Conclusions

The experimental results unequivocally show that both the WaveGAN and DCGAN models failed to effectively learn the task of synthesizing internal combustion engine sounds. However, the failure mode of each architecture was fundamentally different. WaveGAN experienced a catastrophic collapse, marked by an exploding loss function and vanishing gradients, which degraded its output from chaotic noise to complete silence. In contrast, DCGAN entered a state of hidden stagnation and mode collapse; despite an apparently stable loss function, zero gradient norms confirmed that the model was not learning.

Based on these outcomes, Hypothesis H1, which posited that GANs are an effective tool for this synthesis task, must be rejected in the context of this experiment. Neither model generated samples that resembled the original dataset. This finding does not undermine the general potential of GANs but highlights that their effectiveness is critically dependent on achieving training stability through precise hyperparameter tuning and appropriate architectural choices.

Hypothesis H2, stating that the quality of generated samples is predominantly influenced by the input data representation, was fully confirmed. Paradoxically, audio reconstructed from the untrained DCGAN model achieved significantly better quality metrics than samples from the unstable WaveGAN. This indicates that data representation had a decisive influence on the nature of the model's failure. Directly modeling the raw waveform proved to be a highly unstable task, and its failure led to a signal with chaotic properties. The two-stage approach of DCGAN, which simplifies the problem to image generation, was "milder" in its consequences, as the reconstruction algorithm produced a coherent, albeit incorrect, audio signal.

The primary limitations of this study include the failure to achieve training stability, meaning the results primarily document the behavior of the architectures under unsuccessful learning conditions. Other limitations were the use of the Griffin-Lim algorithm, which can introduce artifacts, and the high complexity of the non-stationary signals in the dataset.

6. Summary

This work undertakes the task of investigating and analyzing the capabilities of generative neural networks in the domain of audio synthesis, focusing on a comparison of the effectiveness of two fundamentally different architectures-WaveGAN and DCGAN-in the task of synthesizing internal combustion engine sounds. The key research assumption was the verification of the hypothesis that the input data representation (a one-dimensional raw audio waveform versus a two-dimensional spectrogram image) has a decisive impact on the quality of the generated audio samples.

To achieve the stated objectives, an experimental methodology was employed. Both models were implemented and subjected to a training process on a carefully prepared dataset derived from the Google AudioSet database. The effectiveness of the architectures was evaluated comprehensively, using objective metrics such as Fréchet Audio Distance (FAD), Signal-to-Noise Ratio (SNR), and Fréchet Inception Distance (FID), as well as through subjective perceptual assessment. The entire training process was monitored in detail to analyze its stability.

The experiment demonstrated that both implemented models failed to effectively learn the audio synthesis task, and their training process failed due to vanishing gradients. Despite this, the analysis of the failure modes of both architectures led to a crucial and paradoxical conclusion: data representation is of fundamental importance, determining not only the potential quality of the synthesis but also the very nature of the training failure. The WaveGAN model, operating on the raw signal, experienced a catastrophic failure, generating samples with extremely poor metrics and, ultimately, silence. Surprisingly, the DCGAN model, despite complete stagnation, produced images whose reconstructed audio form achieved significantly better quality metrics. This audio,

although unusable, was statistically closer to a coherent signal than the chaotic noise from WaveGAN. This confirmed the hypothesis that the two-dimensional approach (DCGAN), despite introducing artifacts during the signal reconstruction stage, proved to be more "resilient" to catastrophic collapse under the experimental conditions.

The conducted experiment and its results open avenues for further research that could focus on overcoming the observed limitations. The main directions for potential development include: optimizing the training process to achieve learning stability; replacing the classic Griffin-Lim algorithm with modern neural vocoders (e.g., MelGAN, HiFi-GAN) for audio reconstruction from spectrograms; and exploring hybrid architectures that may offer greater stability and quality in audio synthesis tasks.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, Communications of the ACM, 63(11) (2014) 139-144, https://doi.org/10.1145/3422622.

[2] J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications, IEEE transactions on knowledge and data engineering, 35(4) (2021) 3313-3332, https://doi.org/10.1109/TKDE.2021.3130191.

[3] Documentation of the Google's AudioSet dataset, https://research.google.com/audioset/index.html, [03.2025].

[4] J. W. Kim, R. Bittner, A. Kumar, J. P. Bello, Neural Music Synthesis for Flexible Timbre Control, In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019) 176-180, https://doi.org/10.1109/ICASSP.2019.8683596.

[5] T. Bazin, G. Hadjeres, P. Esling, M. Malt, Spectrogram Inpainting for Interactive Generation of Instrument Sounds, arXiv preprint arXiv:2104.07519 (2021), https://doi.org/10.48550/arXiv.2104.07519.

[6] S. H. Hawley, B. L. Colburn, S. I. Mimilakis, Profiling musical audio processing effects with deep neural networks, The Journal of the Acoustical Society of America 144(3) (2018) 1753, https://doi.org/10.1121/1.5067764.

[7] A. Jabbar, X. Li, B. Omar, A Survey on Generative Adversarial Networks: Variants, Applications and Training, ACM Computing Surveys (CSUR), 54(8) (2021) 1-49, https://doi.org/10.1145/3463475.

[8] J. Berezutskaya, L. Ambrogioni, N. F. Ramsey, M. A. J. van Gerven, Towards Naturalistic Speech Decoding from Intracranial Brain Data, In 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (2022) 3100-3104, https://doi.org/10.1109/EMBC48229.2022.9871301.

[9] C. Donahue, J. McAuley, M. Puckette, Adversarial Audio Synthesis, arXiv preprint arXiv:1802.04208 (2018), https://doi.org/10.48550/arXiv.1802.04208.

[10] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, A. Courville, MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis, arXiv preprint arXiv:1910.06711v3 (2019), https://doi.org/10.48550/arXiv.1910.06711.

[11] J. Engel, K. Krishna Agrawal, S. Chen, I. Gulrajani, C. Donahue, A. Roberts, GANSynth: Adversarial Neural Audio Synthesis, arXiv preprint arXiv:1902.08710 (2019), https://doi.org/10.48550/arXiv.1902.08710.

[12] C. Wang, C. Zeng, J. Chen, O. Xue, HiFi-WaveGAN: Generative Adversarial Network with Auxiliary Spectrogram-Phase Loss for High-Fidelity Singing Voice Generation, In International Symposium on Neural Networks. Singapore: Springer Nature Singapore (2024) 80-92, https://doi.org/10.1007/978-981-97-4399-5_8.

[13] X. Liu, W. Zhang, Z. Zheng, M. Pan, R. Wang, FGP-GAN: Fine-Grained Perception Integrated Generative Adversarial Network for Expressive Mandarin Singing Voice Synthesis, IEEE Transactions on Consumer Electronics, 70(3) (2023) 6054-6063, https://doi.org/10.1109/TCE.2024.3412053.

[14] P. Chandna, M. Blaauw, J. Bonada, E. Gómez, WGANSing: A Multi-Voice Singing Voice Synthesizer Based on the Wasserstein-GAN, In 2019 27th European signal processing conference (EUSIPCO) (2019) 1-5, https://doi.org/10.23919/EUSIPCO.2019.8903099.

[15] J. Nistal, C. Aouameur, I. Velarde, S. Lattner, DrumGAN VST: A Plugin for Drum Sound Analysis/Synthesis with Autoencoding Generative Adversarial Networks, arXiv preprint arXiv:2206.14723 (2022), https://doi.org/10.48550/arXiv.2206.14723.

[16] A. Madhu, K. Suresh, EnvGAN: A GAN-based Augmentation to Improve Environmental Sound Classification, Artificial intelligence review, 55(8) (2022) 6301-6320, https://doi.org/10.1007/s10462-022-10153-0.

[17] S. Liao, S. Lan, A. G. Zachariah, EVA-GAN: Enhanced Various Audio Generation via Scalable Generative Adversarial Networks, arXiv preprint arXiv:2402.00892 (2024), https://doi.org/10.48550/arXiv.2402.00892.

[18] L. Wyse, P. Kamath, C. Gupta, Sound Model Factory: An Integrated System Architecture for Generative Audio Modelling, In International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar) (2022) 308-322, https://doi.org/10.1007/978-3-031-03789-4_20.

[19] S. Andreu, M. Villanueva Aylagas, Neural Synthesis of Sound Effects Using Flow-Based Deep Generative Models, Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 18(1) (2022) 2-9, https://doi.org/10.1609/aiide.v18i1.21941.