Generative AI Framework for Creating a Music Dataset with Quantifiable Acoustic and Cognitive Features

Pol Vierge Meseguer

Abstract

A significant challenge in music cognition research is the scarcity of controlled, unfamiliar musical stimuli, which are essential for experiments where listener familiarity can be a confounding variable. This project addresses this issue by developing a dataset of AI-generated music annotated with acoustic and emotional features. We explored two approaches: (1) building a custom Latent Diffusion Model (LDM), and (2) evaluating five commercial text-to-music systems through subjective and objective analysis. Although our LDM failed to produce usable audio, it provided insight into architectural challenges in modern generative systems. The comparative study—combining listener feedback and Music Information Retrieval (MIR) analysis—identified Suno as the optimal model, offering a strong balance between musical complexity and user appeal. This project establishes a scalable music generation pipeline, laying the groundwork for an annotated dataset that supports experimental research in music perception.

Keywords: Generative AI, Music Cognition, Text-to-Music, Music Dataset, Latent Diffusion Model, Music Information Retrieval

1. INTRODUCTION

Experimental research in music cognition often relies on musical stimuli to probe perceptual, emotional, and neural processes [1][2][3]. A central challenge in designing such experiments is controlling for familiarity: prior exposure to music can significantly alter participants' behavioral and neural responses, making it difficult to isolate the effects of specific acoustic or emotional features. For instance, previous research has shown that simply being familiar with a song can be a crucial factor in emotionally engaging the listener [4]. This "mere exposure effect," where familiarity increases liking ratings, means that any emotional response observed could be due to prior exposure rather than the intrinsic qualities of the music itself. As a result, researchers need unfamiliar, controlled stimuli—yet existing music datasets rarely satisfy these criteria [5].

This constraint has led to the repeated use of a small set of musical excerpts across experiments, limiting the scope and reliability of their studies. Furthermore, existing datasets often lack detailed annotations describing acoustic structure or emotional qualities, making it harder to design stimuli tailored to specific research questions, such as the perceptual difference between major and minor modes or fast and slow tempos.

Recent advances in generative AI, particularly text-to-music models, offer a potential solution. These systems can produce novel, diverse, and prompt-conditioned musical audio—opening new possibilities for generating controlled stimuli on demand. However, it remains unclear whether these models can produce music that is both musically valid and experimentally useful.

To address this gap, we present a framework for generating and evaluating AI-generated music with quantifiable acoustic and emotional features. Our goal is to build a scalable, annotated dataset designed specifically for use in music cognition research.

2. OBJECTIVES

We plan to leverage the capabilities of AI music generation models to produce an extensive library of diverse, high-quality, and free music tracks. This database will be accompanied by detailed metadata (e.g., genre, tempo, emotional valence), which aims to facilitate experimental design and reproducibility. The ultimate goal is to provide neuroscience researchers with a flexible and robust resource that overcomes the current limitations in music cognition research, paving the way for more controlled and varied studies on how the human brain perceives and processes music. In a more detailed manner, the project poses the following objectives:

• Dataset Development: Create an open-access AI-generated music dataset that ensures an structured and accessible option for music-related research along with their respective metadata about the generative model, audio features, and emotional qualities. This involves: Analysing of the physical and musical properties of the generated pieces of music (Signal Processing Analysis). Assessing the emotional qualities of the generated music, such as valence and arousal, conducting a study based on participant evaluation (Emotional Calibration). Publishing the dataset with annotations and analysis results, fostering further research in fields like music cognition (Open Access Contribution).

• Music Generation: Employ state-of-the-art generative models to produce a collection of original songs across different musical properties such as style, key

• Contact E-mail: pol.vierge@autonoma.cat
• Supervised by: Xim Cerdà Company (Department of Computer Science)
• Academic Year 2024/25

or tempo, as well as emotional spectrums. For this, we will be working on the design and implementation of a Text-to-Music model and analysing current commercial solutions for music generation.

• Musical and Emotional Properties Correlation: Study the correlation between the physical, musical, and emotional properties of a piece of audio.

3. STATE OF THE ART

3.1. Emotion-to-Music-Mapping-Atlas (EMMA)

An important precedent in this domain is the Emotion-to-Music-Mapping-Atlas (EMMA) [6], a database that links hundreds of music excerpts to their emotional effects. To capture the nuances of these responses, EMMA employed the Geneva Emotion Music Scale (GEMS), a tool specifically developed for rating musically-evoked emotions. Building on this work, the developers are now creating the AI-EMMAlator, an AI model designed to predict emotional responses from a track's musical features.

3.2. AI Audio Generative Models

Options among Generative Text-to-Audio models are numerous nowadays, from innovative architectures developed and released by research centres to commercial platforms offering their own model's capabilities (e.g., Suno [7]). In this project, we will consider several alternatives, both commercial and non-commercial, as potential engines for our music generation pipeline, as well as a sources of inspiration and benchmarks for developing our own tailored AI music generation model.

3.2.1. Commercial Models

Among the most relevant models and platforms we are considering the following ones as our main music generation engine: Udio [8], Suno [7], Soundraw [9], Aiva [10], and Easy Music [11]. Each of these tools presents distinct strengths in terms of output quality, control over music attributes (e.g., instrumentation, mood), licensing, and technical accessibility. To choose one among them, we performed an analysis on their generated tracks.

3.2.2. MIDI-Based Models

When it comes to AI architectures for music generation, recent advancements in symbolic music modeling have borrowed heavily from natural language processing, treating MIDI data as a language to be understood and generated. One of the most promising approaches in this area is Byte Pair Encoding (BPE) [7], which compresses musical information by identifying common patterns, much like creating contractions in language. By segmenting MIDI event sequences into frequently occurring pairs, BPE creates a compact and expressive musical vocabulary. This compressed representation has been proven to enhance not only the quality of generated music but also the computational efficiency of the Transformer-based models that use it.

Diverging from models focused purely on generation, MusicBERT [12] is designed for a deeper understanding of music. It adapts the BERT architecture, pre-training it on a vast corpus of symbolic MIDI data to model musical structure and hierarchy. By using a specialized BPE tokenization scheme, it is capable of performing downstream tasks like composer identification, genre classification, and analyzing melodic similarity.

3.2.3. Mel-Spectrogram-Based Models

Another approach in music generation models is to operate on Mel-spectrograms, capturing acoustic details while maintaining temporal structure. These models typically encode audio into spectrograms, which are then generated or reconstructed using deep generative models like VAEs, diffusion models, or autoregressive Transformers. Below, we outline some of the most relevant models in this domain:

Riffusion [13] adapts Stable Diffusion, originally designed for Text-to-Image, to work on Mel-spectrograms. The generated spectrograms are then converted into audio using inverse Fourier transformation. Following the use of Natural Language Processing techniques for audio-related tasks, AudioLM [14] models audio as a sequence of discrete tokens across multiple levels of abstraction: semantic tokens for high-level structure and acoustic tokens for fine-grained detail. Originally demonstrated on speech and piano music, this hierarchical approach ensured long-term coherence and high fidelity, laying the essential groundwork for more advanced applications like MusicLM [15].

Building directly on AudioLM's foundation, MusicLM [15] introduced the ability to generate high-quality music directly from text descriptions. It combines pretrained models like MuLan [16] for music-text embedding and SoundStream [17] for high-fidelity waveform synthesis. The result is the current state-of-the-art in its domain: a model capable of producing coherent, multi-minute musical pieces from rich text prompts or even from melodic inputs like humming.

Taking a step further, while many models only focus on interpreting general descriptions, Mustango [18] introduces controllable music generation. Its architecture allows it to follow not just broad text captions, but also precise musical instructions like chords, tempo, or key. This control is achieved through two main components: a Latent Diffusion Model (LDM) that generates the audio, and a Music-Domain-Knowledge-Informed UNet guidance module named MuNet. During generation, MuNet leverages its embedded musical knowledge to ensure the LDM's output adheres to the specific text and musical prompts provided by the user.

3.3. Music Datasets

To develop, fine-tune, and evaluate AI-powered music models, large annotated collections of data are needed. Throughout this process, we will primarily rely on two publicly available music datasets:

The MusicCaps Dataset [15] is a dataset of 5,521 10s music examples from the AudioSet dataset (2,858 from the eval and 2,663 from the train split), each of which is labeled with an English aspect list and a free text caption written by musicians, e.g., "pop, tinny wide hi-hats, mellow piano melody, high pitched female vocal melody, sustained pulsating synth lead", while the caption consists of multiple sentences about the music, e.g., "A low sounding male voice is rapping over a fast paced drums playing a reggaeton beat along with a bass. Something like a guitar is playing the melody along. This recording is of poor audio-quality. In the background a laughter can be noticed. This song may be playing in a bar."

The MusicBench Dataset [18] is an augmented music dataset obtained from altering the MusicCaps dataset along the harmony, tempo, and volume dimensions, as well as caption enhancement. As a result, this dataset consists of ~53K pairs of music audio and description with information on musical attributes like chords, key, and beats.

3.4. MIR (Music Information Retrieval) techniques for Music and Audio Analysis)

Music Information Retrieval (MIR) [19] encompasses a range of computational methods for analyzing, processing, and understanding music and audio signals. Some of them include: Feature Extraction, Automatic Genre Classification, Beat Detection, Melody and Harmony Analysis.

These techniques will be used to evaluate, classify, and annotate the AI-generated music within this project, as they allow for a deep understanding of musical content and are critical for ensuring control over musical features.

4. METHODS

In order to do so, we had considered two different paths with the shared goal of music generation. On one hand, we designed and implemented our own Text-to-Music model taking inspiration from already successful architectures highlighted during the state-of-the-art section (refer to section 3.2). On the other hand, an analysis of current commercial solutions for music generation has been performed in order to select one as our main generative engine.

4.1. Design and Implementation of a Text-to-Music Latent Diffusion Model

4.1.1. Architecture

As outlined in the introduction, one of the objectives of this project is to explore the complexities and challenges of developing a generative model. Following an analysis of current state-of-the-art, we selected a Latent Diffusion Model (LDM) as the architectural foundation. This architecture has been applied in successful models like AudioLM or Mustango, which use a two-stage process for text-conditioned generation. In this project we adopted a similar architectural philosophy:

In this approach we used the MusicBench dataset to first train a Variational AutoEncoder (VAE) to compress high-dimensional data—in this case, Mel-spectrograms—into a small and efficient latent space, and to reconstruct the original data from these latent representations. Next, a UNet was trained to perform a conditional denoising task within the latent space obtained through the encoder of the VAE. For inference (generating music from a new prompt), we start with a tensor of pure random noise. Guided by the text prompt, the UNet iteratively denoises this tensor until it becomes a clean latent representation. Finally, this is passed through the VAE's decoder one time to generate the final spectrogram.

This two-stage process allows breaking a complex task into two more manageable parts: Representation Learning (VAE) and Conditional Generation (UNet). However, the quality of the conditional generation is fundamentally dependent on the representation learning, or in other words: the autoencoder dictates the maximum possible quality of the final output. The autoencoder must create a latent representation that not only compresses the data but also preserves the essential perceptual information of the original spectrogram. If it fails to do so, the architecture will experience a loss of information during the compression stage, that is, an "information bottleneck" that no subsequent component, including the diffusion model, can overcome. And at the same time, the reconstruction quality of the decoder defines the ceiling for the audio fidelity of the entire model. Therefore, the initial and most critical phase of this implementation was the development of the autoencoder.

4.1.1.1. Variational AutoEncoder

As base architecture, the first implementation was a standard convolutional autoencoder to explore its capabilities and limitations. The encoder was designed with a series of convolutional layers to progressively downsample the input Mel-spectrogram into a compact latent vector, and a corresponding set of transposed convolutional layers to decode it back to its original dimensions.

A more sophisticated approach is the use of Generative Adversarial Networks (GANs) [20]. These do not rely on a fixed, mathematical error metric. Unlike standard autoencoders, they employ a second neural network—the discriminator—which learns to differentiate between real and generated data. This adversarial process forces the generator to learn to model the complex statistical distribution of the training data itself to produce outputs that are not just mathematically, but also perceptually close to the training data. In this revised model, the original autoencoder was repurposed as the GAN's Generator (G), and a second network, a patch-based convolutional classifier (PatchGAN)–the Discriminator (D)–, was introduced.

4.1.1.2. Denoising Diffusion Probabilistic Model

The generative core of the Latent Diffusion Model (LDM) is the conditional diffusion model: a Denoising Diffusion Probabilistic Model (DDPM), which learns to create data by reversing a process of gradually adding noise. It involves two main stages: a forward process and a learned reverse process.

The forward process is fixed. It takes a clean data sample—in our case, a latent representation from the VAE—and slowly adds random noise to it over a series of timesteps so that at the end of this process, the original sample is indistinguishable from pure noise. The reverse process is where the learning happens. Here, a neural network is trained to reverse the noising process. Starting with a tensor of pure random noise, the model learns to gradually remove the noise step-by-step, ultimately arriving at a clean latent representation. The neural network responsible for this task (reverse process) is a UNet, chosen because of its effectiveness when processing image-like data, which is what the latent representations of Mel-spectrograms are. Its structure consists of a symmetric encoder-decoder path with skip connections connected by a central bottleneck.

The key feature of the UNet is the skip connections that link layers from the encoder directly to corresponding layers in the decoder. These connections create a shortcut for fine-grained details from the early stages to be passed to the later stages. In our case, audio generation, this helps ensure that precise temporal and frequency information is not lost during the downsampling-upsampling process.

The current definition of our generator allowed for unguided generation of samples from the learned data distribution, but offered no user control. To generate music from text, the denoising process must be conditioned on the meaning of a text prompt. We achieved this by integrating text information into the UNet using a cross-attention mechanism, an approach used in controllable models like Mustango. First, the text prompt (e.g., "A mellow piano melody with a pop beat") is converted into a numerical representation using a pre-trained text encoder. Following the approach of models like MusicLM, we use MuLan, a model pre-trained on music and text data, to capture the semantic meaning of the prompt. Then, the text embedding is injected into the UNet to guide the denoising. This is done with cross-attention layers placed at specific layers of the UNet alongside self-attention blocks. This mechanism allows the UNet to "look at" the features of the text prompt and determine which parts of the prompt are most relevant at that moment. For example, it might focus on "mellow piano" when generating harmonic content and "pop beat" when generating rhythmic structure, steering the denoising process toward a final latent representation that matches the user's description.

The UNet must adapt its behavior depending on the current step in the denoising process. Early on, when the input is mostly noise, its task is to define the broad musical structure. Later, when the input is already structured, it must focus on refining details. To handle this, the model needs to know the current timestep, t. We provide this context by converting the timestep number into a high-dimensional vector using sinusoidal positional embeddings, this way each step has a unique embedding. This time embedding is then fed into each ResNet block of the UNet, ensuring the entire network is aware of its position in the generation process and can adjust its function accordingly.

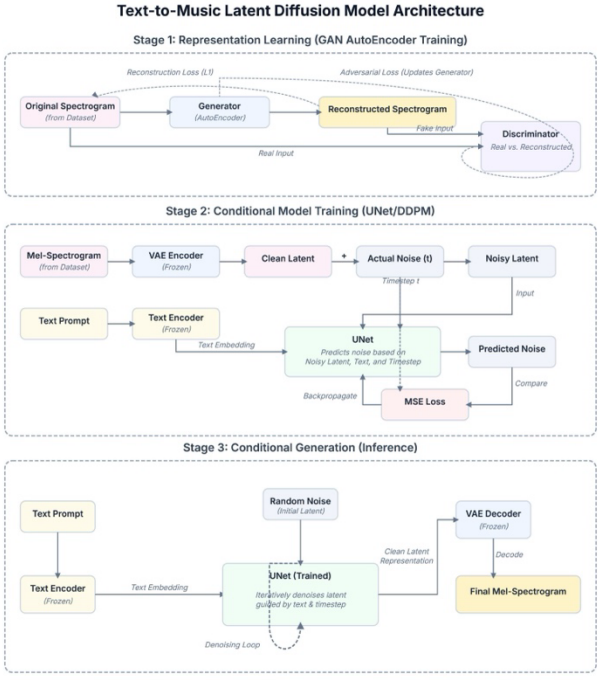

Fig. 1 Model training and inference pipeline. Stage 1 (Up): GAN AutoEncoder training. Stage 2 (Centre): UNet training. Stage 3 (down): Inference.

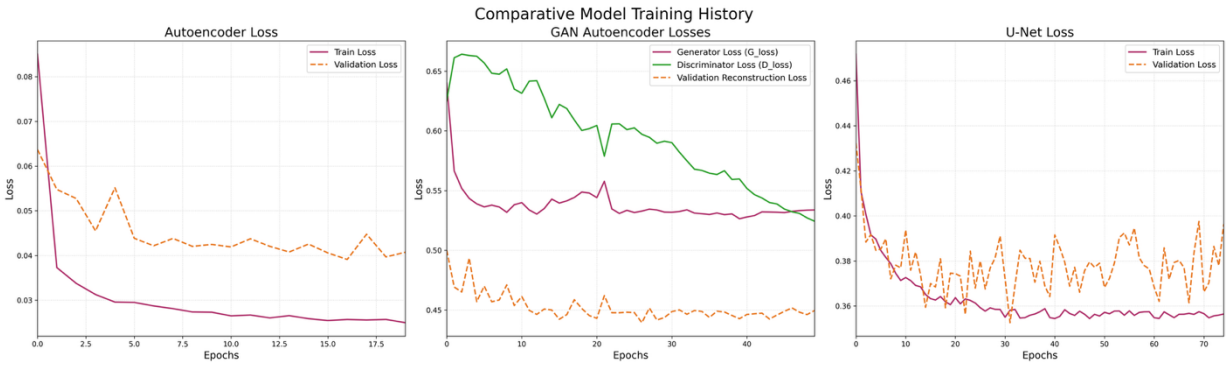

Fig. 2. Training and validation losses accross different models. Left side: AutoEncoder training and validation losses (MSE + L1). Right side: UNet training and validation loss (MSE). Centre: generator training loss (adversarial + 100*L1, divided by 10 for visual comparison), discriminator training loss (adversarial), and generator validation loss (L1, multiplied by 10 for visual comparison).

4.1.2. Training

The implementation and training of these models have been done in Pytorch [21] and run in a Linux Ubuntu virtual machine with a NVIDIA GeForce GTX TITAN X (8 cores, 32GB of RAM, 300GB of disk and a GPU of 12GB).

4.1.2.1. VAE training

To train the AutoEncoder, a hybrid loss function combining L1 (Mean Absolute Error) and L2 (Mean Squared Error) was selected, which is a standard and widely adopted baseline for training image and signal reconstruction models.

To train the GAN model, a hybrid loss function was implemented, combining the strengths of two different metrics: a direct reconstruction loss (L1) and an adversarial loss from the GAN framework. The adversarial loss component is what drives the model to produce realistic outputs. The Discriminator is trained to become an expert at telling "real" spectrograms (from the dataset) apart from "fake" ones (from the Generator). In turn, the Generator's goal is to fool the Discriminator. This dynamic treats the discriminator as a learned, perceptual loss function that assesses the realism of the output and forces the Generator to avoid producing blurry or unrealistic representations.

However, relying solely on adversarial loss can make training unstable and may not guarantee that the output is a faithful reconstruction of the input. To solve this, a traditional L1 (Mean Absolute Error) loss was added to the Generator's loss. This loss directly measures the element-wise difference between the original spectrogram and the generated one, ensuring that the model is learning to reconstruct the content accurately, not just generate an arbitrary, albeit realistic, sample.

4.1.2.2. DDPM training

For the Denoising Diffusion Probabilistic Model (DDPM), we trained the UNet to predict only the noise that was added at each timestep. The training goal is to minimize the difference (Mean Squared Error) between the noise predicted by the model and the actual noise that was added in the forward process. All in all, we are reframing the generation task as a denoising problem. During the DDPM training a spectrogram sample is encoded into a clean latent representation using the frozen VAE encoder and its caption is encoded into a conditioning embedding using the frozen text encoder. A random timestep is chosen, and the corresponding amount of noise is added to the clean latent representation, creating a noisy latent. Then the UNet takes this noisy latent, the timestep embedding, and the text conditioning embedding as input and predicts the noise that was added. After that, the Mean Squared Error (MSE) loss is calculated between the predicted noise and the actual noise, and the backpropagated to update the weights of the UNet model.

4.1.3. Inference

The inference pipeline is how new music is generated from a text prompt. In this process we first start with a tensor of random noise (with the same dimensions as the latent space) and a text prompt, which is converted into a conditioning vector by the text encoder. Then an iterative loop runs backward from the last timestep to the first. In each step, the UNet predicts the noise in the current latent tensor, and a scheduler algorithm uses this prediction to compute a slightly less noisy version for the next step (denoising loop). Once the loop finishes, the result is a clean latent representation that reflects the text prompt. This latent is passed through the VAE's decoder once to generate the final Mel-spectrogram.

4.2. Commercial Models for Music Generation

We selected five commercial models as possible candidates to perform song generation for our dataset: Suno, Udio, EasyMusic, Soundraw and Aiva. Assessing music quality is a complex task given its

subjective nature. In order to decide on a generative engine, we performed two different analysis on generated tracks: Questionnaire Assessment (likeness, curiosity, perceived complexity, human-machine perception, and genre classification for prompt accuracy), and Music Information Retrieval (MIR) analysis of physical features.

4.2.1. Subjective Assessment

4.2.1.1. Participants

The subjective assessment was performed conducting an experiment based on participant criteria, where 20 people participated. The mean age was of 34 years old (std = 16.2, min = 21, max = 59), 12 were female (60%), 7 male (35%), and 1 non-binary (5%). When it comes to music education, 5 had no musical knowledge (25%), 9 reached a basic level in high school (45%), 5 reached an intermediate level through introductory music studies or local music schools (25%), and 1 of them had knowledge compared to university studies or professional conservatory (5%).

4.2.1.2. Stimuli and procedure

For each one of the models, 10 instrumental tracks were generated, each one embodying one of the following genres: Pop, Rock, Jazz, Classical, Rythym and Blues (R&B), Heavy Metal, Electronic Dance Music (EDM), Indie, Folk, and Hip-Hop. All models, except Soundraw, work with text prompt. In these cases, the prompt consisted solely on the word "instrumental" followed by the genre. In Soundraw, generation is done based on genre, mood, theme, tempo, and instruments. Genres are limited, so when working with this model, if the genre was available the prompt was just its selection, on the contrary, a combination of genres and mood was done to depict the desired genre. To avoid bias on the track selection, the first generated track for each prompt was the one used for this analysis in each model.

The questionnaire was programmed using Google Script along with Google Froms integration and the experiment has conducted online. In this form, participants listened to 30s-excerpts from all the generated songs (50 songs) in random order and were asked to answer the following questions: *How much did you like the song?* (liking), *How much do you agree with the following statement: This piece was created by a human* (perceived humanity), *How complex is the song?* (complexity), *How curious do you feel about continuing to listen to the song?* (curiosity), *Which genre(s) would you classify this piece under?* (genre classification for prompt accuracy). All questions were answered using a 1-7 scale, except the genre question which was a multiple choice question with all 10 genres as possible answers.The 30s of each song were selected manually based on their most promising part, trying to keep neutrality between all of them.

4.2.2. Acoustic Features Analysis

The analysis of physical properties of the 50 generated music pieces was conducted using the MIR ToolBox [19] in MATLAB R2024b (MathWorks, USA.) [22]. Following a similar approach as in [1], we computed a total of 21 features for each song, which could be divided into six different domains: dynamics, timbre, harmony, rhythm, articulation and structure. After computing them, the mean values for each feature and model were computed, and one-way ANOVA was performed in R [23] to obtain the significant features when it came to differences between models, we will analyse these futures in the results section.

TABLE 1. Significant MIR ToolBox Acoustic Features

Musical Domain	Acoustic Feature	Perceptual Meaning	Time Window
Dynamics	RMS	Loudness	50 ms (50%)
	Low Energy	Loudness contrast	50 ms (50%)
Timbre	Roughness	Dissonance	50 ms (50%)
	Spectral Flux	Timbral change over time	50 ms (50%)
Harmony	HCDF	Harmonic richness	743 ms (10%)
Rhythm	Tempo	Tempo	3 s (10%)
Articulation	Attack Time	Sharpness or Percussiveness	3 s (10%)
Structure	Novelty	Musical contrast	1.6 s

Note. The numbers in parentheses indicate the overlap rates. Novelty was computed for the entire piece of music and its time window depends on kernel size (64).

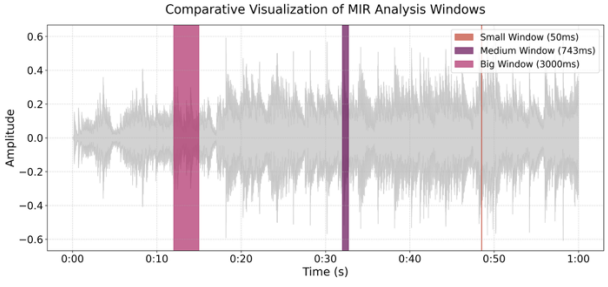

Fig. 3 MIR Windows size comparison on a 1min-excerpt from a song. 50ms small window (right side, orange), 743ms medium window (centre, purple), 3s big window (left side, pink).

4.2.3. Music Generation Pipeline

After the selection of a model as the primary generative engine, a generation pipeline was designed and implemented to create the tracks of synthetic music dataset. This process was structured to ensure a diverse and controlled generation of musical tracks, each linked to specific, descriptive metadata.

The pipeline consists of two main stages: dynamic prompt generation and an automated submission-and-retrieval workflow. The entire process was automated using Python scripts, which handle the communication with an API, manage job submissions, and organize the resulting audio files and metadata.

4.2.3.1. Prompt Design and Generation

A prompt generation system was designed as a structured combinatorial prompting so that prompts were consistent, controllable, and scallable in terms of possible combinations. Each prompt was designed to be for an instrumental track and was built from a fixed structure combining a primary genre with three key descriptive components: dynamics (loudness), timbre, and emotion.

The final prompt structure is as follows: "An instrumental {emotion} {genre} track, featuring {timbre} and a {dynamic} character." An example of a generated prompt is: "An instrumental sad Pop track, featuring a dark sound and a soft character." In this example, "Pop" is the genre, while "sad," "dark," and "soft" are the descriptors for emotion, timbre, and dynamics, respectively. These descriptive elements were the only variables that changed between the generated songs.

The list of genres was compiled by researching principal music genres, resulting in a selection of 14 distinct styles: Classical, Experimental, Blues, Country, Easy Listening, Electronic, Folk, Hip Hop, Jazz, Pop, R&B and Soul, Rock, Metal, and Punk. The descriptors for dynamics, timbre, and emotion were carefully selected from a specialized lexicon designed for describing sound presented on a publication aimed at creating a common vocabulary for audio description [24].

4.2.3.2. Generation Process

The process of generating music and retrieving the final audio files was automated through a series of Python scripts that interacted with the API. When a prompt is generated using the structure described previously, is sent as an instrumental generation job to the API. As this task is not an instantaneous process, to manage this, a unique URL, provided by a service called Webhook.site, is included in the API request as a callBackUrl. Once the model has finished generating the audio, it sends a notification (a callback) to this URL, this way multiple jobs can be submitted without having to wait for each one to complete individually. Simultaneously, another script continuously monitors the callbacks received at the webhook URL. When a "complete" notification arrives, it contains the necessary information to retrieve the generated audios (commercial models tend to create two songs per prompt) and records all relevant information into a CSV file. This log file links each audio file to its corresponding generation data, including a timestamp, the full text of the prompt used, the assigned title, the specific descriptors (genre, emotion, timbre, dynamics), and the unique track and task IDs provided by the API.

5. RESULTS

5.1. Implemented Text-to-Music Latent Diffusion Model

When it comes to the basic AutoEncoder architecture, despite the widespread use of L1 and MSE losses and numerous experiments involving adjustments to the model's depth, the use of ResNet blocks, and variations in the training data volume (using a 25% subset of the MusicBench dataset for initial tests), and even switching to STFT loss, the reconstructions produced by this model were consistently suboptimal.

Although training converged (Figure 2), reconstructions consistently exhibited severe blurring, smoothing, and loss of high-frequency content (Figure 4), which translate to noise-prominent and muffled sounds when converted into an audio waveform.

This is not a minor artifact or a simple failure of model capacity, is a well-documented issue of using pixel-wise loss functions – like L1 and MSE – for complex, structured data like images or spectrograms [25]. Both L1 and L2 losses operate by independently penalizing the error for each individual bin in the

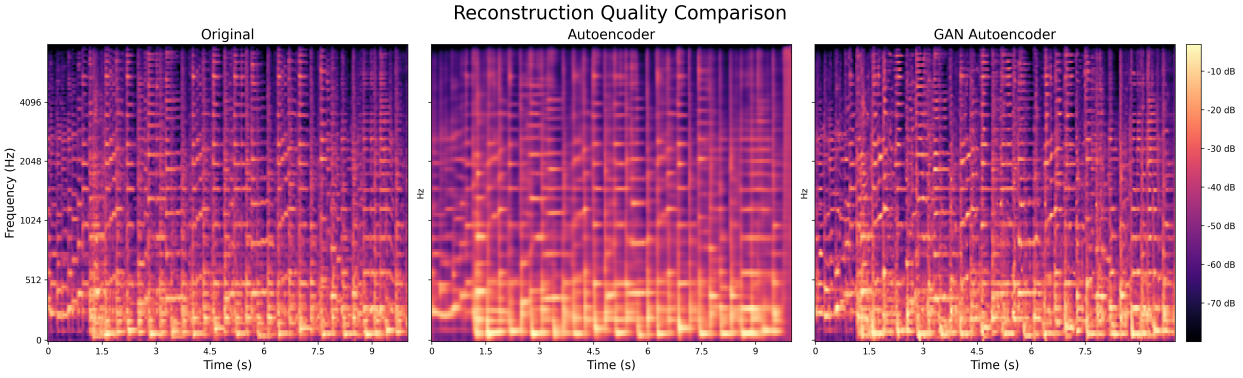

Fig. 4 AutoEncoders reconstruction comparison. Left side: Original mel-spectrogram of a 10s test audio sample. Centre: AutoEncoder reconstruction. Right side: GAN AutoEncoder reconstruction.

spectrogram, which can lead to failure when it comes to account for spatial relationships and structures, which are essential in the audio domain, where these define perceptually important features like harmonics, transients, and textures.

Moreover, this kind of loss functions struggle with the multi-modal nature of audio data. For any given low-frequency structure in a spectrogram, there are many plausible high-frequency details that could accompany it; there are multiple valid reconstructions. When this happens, the model learns to produce a 'safe', averaged compromise, erasing the high-frequency details, which are essential for audio clarity.

To mitigate this, we replaced the decoder with a GAN-based generator (PatchGAN discriminator), whose reconstructions capture the essential macrostructure, including the primary harmonic content, while introducing minimal noise or artifacts (Figure 4). However, looking at the difference between the reconstructed and the original melspectrogram (Figure 5), we can identify that there is a loss of fine detail, particularly in the higher frequency ranges (the reddish tint in the upper frequencies), and a slight softening of sharp transient events (the strong red vertical lines). Furthermore, we can observe evidence of temporal smearing, where the energy of sharp onsets is spread out, appearing as red lines (underestimated peak) followed by blue (overestimated shoulder) regions. Perceptually, this translates to a slightly noised, smoothed, and less dynamic version of the original audio. All this suggests persistent limitations in the learned latent space: it compresses data but fails to preserve key perceptual features.

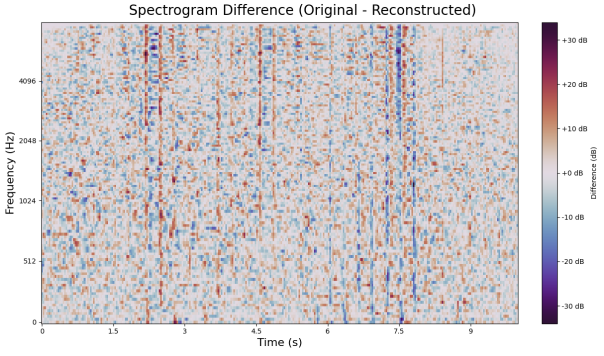

Fig. 5 Difference between the original and the reconstructed spectrogram by the GAN AutoEncoder model.

The Denoising Diffusion Probabilistic Model training loss curves showed stable convergence (Figure 2), and the model produces structurally coherent spectrograms, and most importantly, different from each other in ways that directly correspond to the prompts, showing some level of semantic conditioning (see Figure 6). Unfortunately, there is no timbric or harmonic definition, resulting in audios that resemble structured noise rather than coherent music.

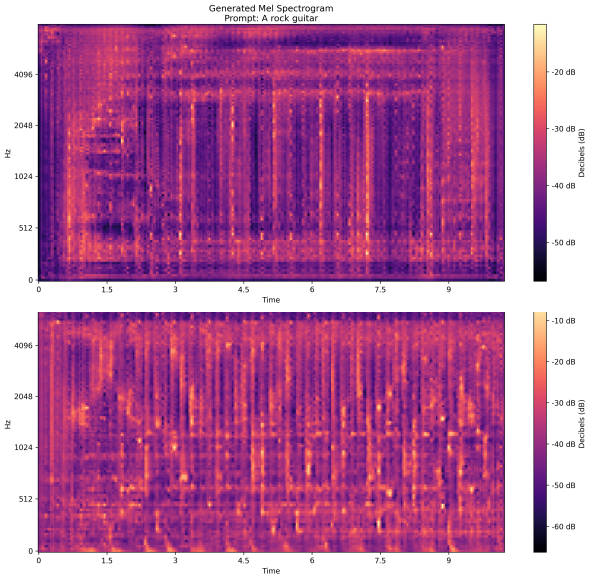

Fig. 6 Audio samples generated by the trained Latent Diffusion Model conditioned by the "A pop piano" (down) and "A rock guitar" (up) prompts.

5.2. Commercial Models for Music Generation

5.2.1. Subjective Assessment

When observing the mean results of the subjective assessment (Figure 7), a clear trend emerged: across the general categories of liking, perceived humanity, complexity, and curiosity, EasyMusic consistently achieved the highest scores, with Suno and Udio following as close competitors. Results regarding genre classification for prompt accuracy assessment are commented on Appendix A.1 as they were shown to be highly conditioned by the listeners knowledge and familiarity with each of the genres.

To understand these results more deeply, we performed a one-way ANOVA, which revealed significant differences between models across eight distinct genres:

Classical ($p < 0.001$), Jazz ($p < 0.001$), Indie ($p < 0.001$), Hip-Hop ($p < 0.001$), Pop ($p = 0.002$), Rock ($p = 0.002$), Heavy Metal ($p < 0.001$), and R&B ($p < 0.001$).

5.2.2. Acoustic Features Analysis

After computing the MIR acoustic features of the 50 songs generated for the commercial models' comparison and analysis, 8 were found to show a significant difference between models (Figure 8):

Root Mean Square (RMS) ($p < 0.001$), LowEnergy ($p=0.043$), SpectralFlux ($p=0.002$), Roughness ($p < 0.001$), Harmonic Change Detection Function (HCDF)

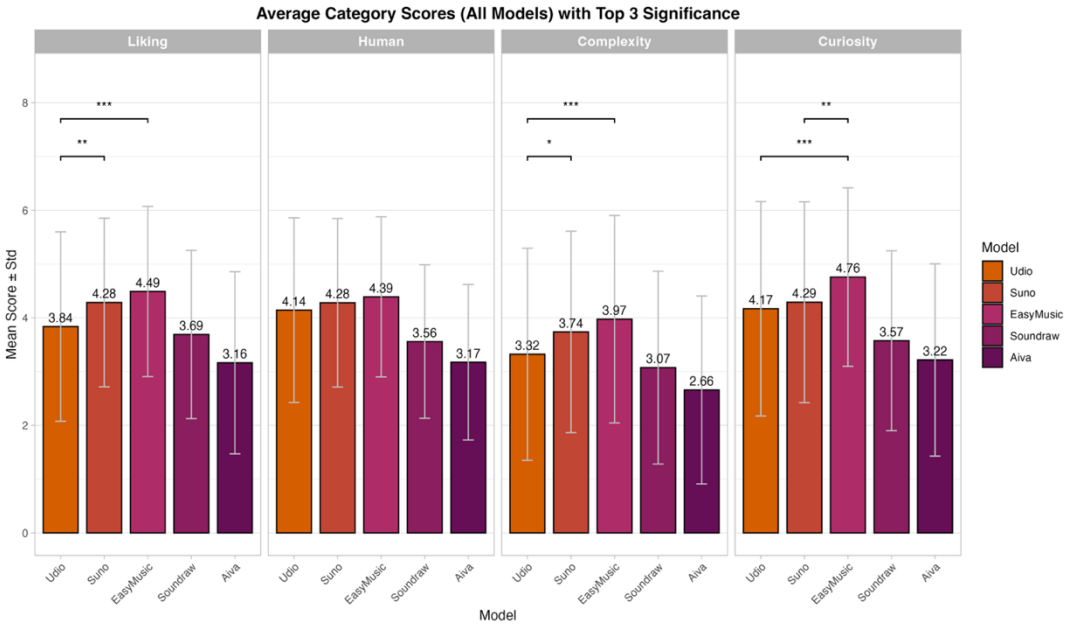

Fig. 7 Average scores for liking, perceived humanity (human), complexity and curiosity facets across all five models. Black bars in the upper part of the plots represent significance between model pairs (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$). This significance has only been computed for Udio, Suno and EasyMusic (Top 3 models).

(p=0.029), Tempo (p=0.028), AttackTime (p=0.020), and Novelty (p=0.019).

RMS and Low Energy are dynamic indicators, more specfically, they reflect the perceived loudness and the presence of quieter sections respectively. Easy Music exhibited the highest RMS energy and lowest Low Energy values, suggesting dynamically compressed and/or consistently energetic outputs. In contrast, Aiva and Soundraw had the lowest RMS and highest Low Energy values, which mean quieter sounds. Suno and Udio seemed to achieve better balance between these two features. However, Suno's RMS values are more consistent within tracks, meanwhile Udio's may vary depending on the output.

Spectral Flux and Roughness are timbirc features, roughness is related to dissonance, meanwhile spectral flux indicates timbral change over time. Udio had the highest Spectral Flux and Roughness values, indicating frequent changes in the spectral content and more complex or harsh textures. Aiva and Soundraw displayed the lowest Roughness, which may correspond to smoother, cleaner or simpler sounds. Easy Music showed intermediate values, suggesting a balance between them.

Harmonic analysis, based on Harmonic Change Detection Function (HCDF), related to harmonic richness, revealed that Suno and Udio had the highest mean HCDF values, which translates to more frequent or pronounced harmonic changes. In contrast, Aiva and Soundraw scored lower, indicating more static harmony.

Tempo analysis showed similar values on most models, except on Aiva, which showed to lead to slower-paced music compared to the other models.

The attack refers to the initial part of a sound, in this case, attack time refers to how quickly a sound reaches its maximum amplitude after being triggered. A fast attack time makes a sound sharper and more prominent, usually related to percussiveness, while a slower attack time makes the sound softer and smoother. Udio had the longest average attack times, and Easy Music had the shortest.

Finally, novelty measures how much change occurs over time, so the higher the novelty, the more musical contrast and changes, while the lower the novelty, the more repetitive or static strucure is. Novely was highest in Udio and Aiva, suggesting these models generate music with more varied structures. Suno and Easy Music showed lower values, implying more repetitive or homogenous compositions.

Treating genres separately (see figure in Appendix A.2) helped linking some of these acousitc atttributes to the models' performances in the subjective assessment. For instance, EasyMusic's lead in genres like Hip-Hop and Heavy Metal could be explained by its high-energy and percussive profile, ideal for this type of music. In other genres, however, both Suno and Udio delivered tracks that were rated as highly, if not more so, than EasyMusic's. For example, in Jazz, a harmonically complex genre that could be directly linked to their high HCDF scores compared to EasyMusic.

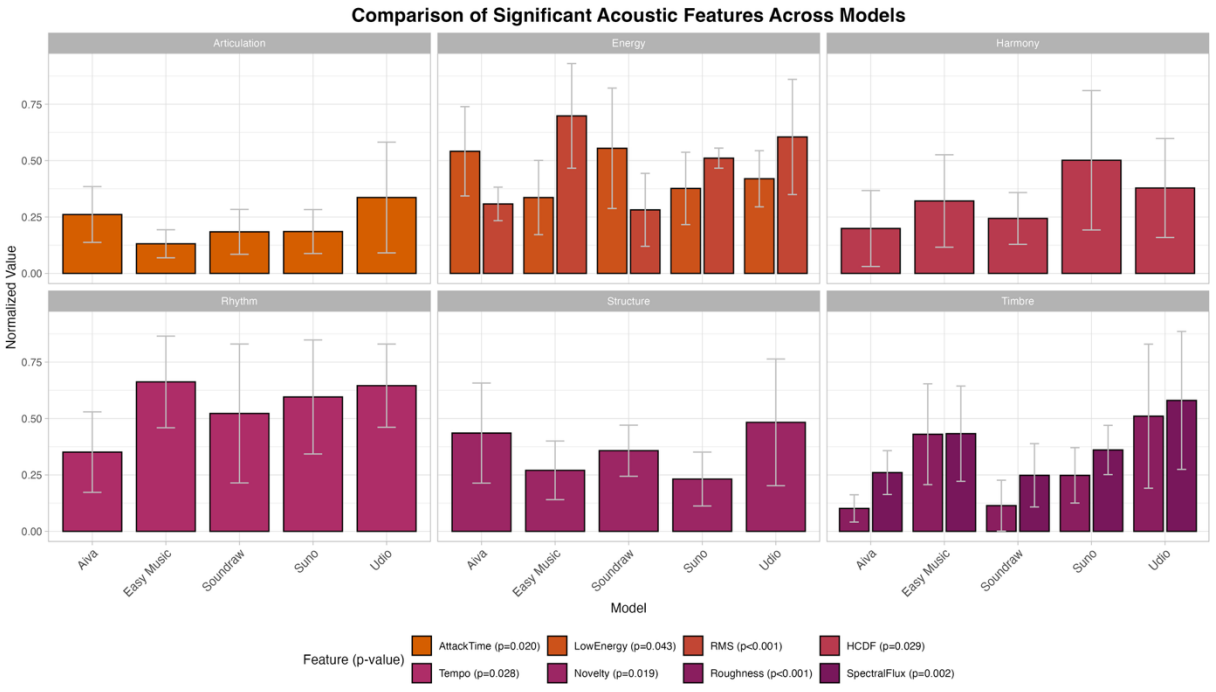

Fig. 8 Mean values of the acoustic features showing significant difference between the five commercial models. Features are divided into musical domains (Articulation, Energy, Harmony, Rythym, Structure and Timbre). Grey error bars correspond to standard deviation.

6. DISCUSSION

The results clearly indicate that our self-implemented LDM, in its current form, is not suitable for music dataset generation. While it helped deepen our understanding of the components used in state-of-the-art architectures (e.g., MusicLM, Mustango), the model failed to synthesize musically intelligible output. This failure is likely due to information bottlenecks in the latent space, inherited from the VAE. Even with GAN-based enhancements, the encoder failed to preserve fine-grained structure required for plausible synthesis. Since the diffusion model operates exclusively in this compromised space, its output is fundamentally constrained. The use of simple auto-encoding techniques—even with adversarial training—was not sufficient. More advanced strategies such as architecture innovation implementing VQ-VAEs or transformer-based encoders, or perceptual similarity losses like SSIM or STFT may be needed for future iterations. Given this, we discard the possibility of using this model (at least, at its current configuration) to generate music tracks for the dataset and will rely on commercial models to perfrom such task.

On the commercial side, the combined results of the objective and subjective analyses did not yield a singular "best" model, as it depends on whether the primary goal is to produce psychoacoustically pleasing audio or musicologically accurate and complex compositions. Yet, it highlighted three strong candidates: Udio, EasyMusic and Suno.

EasyMusic's success in the subjective ratings can be attributed to a profile optimized for immediate psychoacoustic appeal. Its high RMS, low Low-Energy, and short attack times create loud, punchy, and dynamically consistent outputs. Specifically, this loudness aligns with the production values of modern commercial music, creating a sound that is familiar, appealing, and requires minimal effort to process. Its lower novelty score indicates structural predictability, which can also contribute to ease of listening and general appeal for a passive listener.

Conversely, Udio's profile, while leading to the highest genre accuracy, seems to be less optimized for this kind of immediate gratification. Its high HCDF, Novelty, and Spectral Flux scores point to music that is harmonically, structurally, and timbrally complex. This is positive when it comes musicological integrity, but it may be hindering its perception by untrained ears.

Finally, Suno emerges from the analysis as a compelling third option that attempts to bridge the gap between these two extremes. On one hand, it scored highly in the subjective ratings, being the second highest model and having no significant difference between with EasyMusic except in curiosity. On the other, it demonstrated significant musicological strength, matching Udio with the highest mean HCDF values, which points to a sophisticated ability

to generate complex and evolving harmonic structures. Although having its own trade-offs, like novelty, Suno's balance of psychoacoustic appeal with harmonic richness makes it the most viable option for building our experimental dataset.

It is worth mentioning that there is a discrepancy arose from the "Complexity" rating. Although participants rated EasyMusic's output as the most complex, the objective MIR analysis shows that Udio and Suno generate music with far greater harmonic (HCDF) and structural (Novelty in the case of Udio) complexity. This suggests that the participants may have interpreted "complexity" not in a musicological sense, but as a measure of sonic density or "fullness", which could be related to EasyMusic's RMS values. Its dynamically compressed, "wall of sound" could be perceived as "complex" by an ear not trained to parse harmonic progressions or musical structures. This divergence between subjective perception and objective acoustic metrics could be the focus of future studies.

7. CURRENT STATE OF THE PROJECT AND FUTURE STEPS

With the selection of an AI engine and the pipeline to perform massive generation, we now have obtained 957 complete music tracks, and their acoustic features are being computed following the same approach as the one presented in this report.

Next steps involve the analysis of these acoustic features and the design and conducting of an experiment to collect cognition-related data of these songs. After requesting funding, we plan on recruiting between 375-400 participants to assess these songs in terms of liking, valance, arousal, curiosity and complexity. With these two domains (acoustics and cogntion) covered, we can then annotate the dataset and prepare it for its open access publication. In addition, we plan to perform a study on emotion to explore the correlation between physical/musical features and their evocked emotion on people, as well as how do machines differ with humans when it comes to relating such features to a given emotion.

8. ETHICAL IMPLICATIONS

This project leverages commercial platforms, some of them which remain largely opaque about the specific composition of their training data. This not only happens with commercial models, but also with publicly available datasets. The easy access to these technologies and the lack of regulation regarding them create a false sense of ethical and legal security. As a result, when a researcher uses this kind of technologies, the resulting outcome, while intended for the public good of scientific research, as it is our dataset or LDM, could be unknowingly benefiting on the mass, uncompensated, and often non-consensual ingestion of pre-existing human art. At the same time, these platforms and datatsets are most probably biased towards western-predominant popular music. When training new models with this data and using already existing models trained similarly, we are exposed to a significant risk of cultural homogenization, where diverse, experimental, and non-western musical traditions are marginalized. To mitigate this bias in our data, we added a component in our prompt generation pipeline to add an ethnic factor and therefore, diversify the types of music being generated. This, however, was discarded due to the lack of transparency and probable bias regarding the training data used in the commercial models we were leveraging — if they were trained mostly with western music, they will be most likely not be capable to generate non-western music.

When it comes to AI, transparency is paramount. AI developers, both commercial and academic, must be open about the data used to train their models. We, as researchers, are at times limited to using these tools, but must, in turn, be transparent about their methodologies and the nature of their AI-generated outputs. We frame project inside the paradigm of Human-Centered AI (HCAI), which positions AI as an assistive tool to augment and enhance human creativity, rather than replace it. While it uses generative tools, its end product is not intended for mass-market competition with human artists. Instead, we aim to create a specialized scientific instrument designed to overcome current challenges and answer specific questions in the field of music cognition.

9. LIMITATIONS

The project has faced some limitations. For the AI implementation part, these have been mainly related to time and computational resources. The training of these models is very time consuming and computational demanding, which has restricted the research capacity when testing new architectures and configurations. The subjective assessment study for the commercial models has been limited mainly in terms of participant sample size and controllability during the experiment as it was conducted online. Recruiting has been voluntary, and to make the experiment realistically plausible and ensure a minimum amount of participation the listening and questionnaire answering could not exceed the hour. This led to a reduced amount of 10 songs per model and a final sample size of 20 people, which may not be sufficient to extrapolate to decisive conclusions.

10. CONCLUSIONS

In this project, we have explored the possibility of creating an AI model to generate prompt-conditioned songs for the creation of a music dataset, as well as the potential of using already-existing commercial solutions for such task. Although the LDM implementation has not been successfull, it has served as a way of

understanding what is behind some of state-of-the-art architectures. The study for the selection of a commercial model has provided insights about the individual strengths and weaknesses of the selected engines, and have highlighted Udio, EasyMusic and Suno as the most reliable to perform music generation. Finally, this project is just a precedent; this work has laid down the first brick in the creation of an annotated dataset of synthetic music for music cognition studies.

ACKNOWLEDGEMENTS

This work was supported by the project Cátedra ENIA UAB-Cruïlla (TSI-100929-2023-2) and the MINECO project (PID2023-151083NA-I00).

CODE AVAILABILITY

The source code associated with this study is publicly available on GitHub at the following repository: https://github.com/polvime/GenAIDataset

REFERENCES

[1] K. Mori, "Decoding peak emotional responses to music from computational acoustic and lyrical features," *Cognition*, vol. 222, May 2022, doi: 10.1016/j.cognition.2021.105010.

[2] "Intensely pleasurable responses to musi...gions implicated in reward and emotion".

[3] L. Ferreri *et al.*, "Dopamine modulations of reward-driven music memory consolidation," *Ann N Y Acad Sci*, vol. 1502, no. 1, pp. 85–98, 2021, doi: 10.1111/nyas.14656.

[4] C. S. Pereira, J. Teixeira, P. Figueiredo, J. Xavier, S. L. Castro, and E. Brattico, "Music and Emotions in the Brain: Familiarity Matters," *PLoS One*, vol. 6, no. 11, 2011, doi: 10.1371/journal.pone.0027241.

[5] G. Cardona, X. Cerda-Company, E. Sozza, D. Cucurell, L. Ferreri, and A. Rodríguez-Fornells, "Music-Driven Curiosity: Exploring Its Role in Information-Seeking and Memory in Laboratory and Real-Life Context." 2025. Submitted for publication.

[6] H. Strauss *et al.*, "The Emotion-to-Music Mapping Atlas (EMMA): A systematically organized online database of emotionally evocative music excerpts," *Behav Res Methods*, vol. 56, no. 4, pp. 3560–3577, Apr. 2024, doi: 10.3758/s13428-024-02336-0.

[7] Suno AI, "Suno." [Online]. Available: https://suno.com/

[8] Udio, "Udio AI Music Generator." [Online]. Available: https://www.udio.com/

[9] SOUNDRAW, "AI Music Generator SOUNDRAW." [Online]. Available: https://soundraw.io/

[10] Aiva, "AIVA, the AI Music Generation Assistant." [Online]. Available: https://www.aiva.ai/

[11] EasyMusic, "EasyMusic AI Music Generator." [Online]. Available: https://easymusic.ai/

[12] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training," Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.05630

[13] S. Forsgren and H. Martiros, "Riffusion - Stable diffusion for real-time music generation ." [Online]. Available: riffusion.com

[14] Z. Borsos *et al.*, "AudioLM: a Language Modeling Approach to Audio Generation," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.03143

[15] A. Agostinelli *et al.*, "MusicLM: Generating Music From Text," Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.11325

[16] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "MuLan: A Joint Embedding of Music Audio and Natural Language," Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.12415

[17] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.03312

[18] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, "Mustango: Toward Controllable Text-to-Music Generation," Nov. 2023, [Online]. Available: http://arxiv.org/abs/2311.08355

[19] O. Lartillot, P. Toiviainen, and T. Eerola, "A Matlab Toolbox for Music Information Retrieval," in *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization*, C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, Eds., Berlin, Heidelberg: Springer, 2008.

[20] I. J. Goodfellow *et al.*, "Generative Adversarial Nets." [Online]. Available: http://www.github.com/goodfeli/adversarial

[21] P. Team, "PyTorch." [Online]. Available: https://pytorch.org

[22] The MathWorks Inc., "MATLAB version: 24.2.0 (R2024b)," 2022, *The MathWorks Inc., Natick, Massachusetts, United States*. [Online]. Available: https://www.mathworks.com

[23] Posit team, "RStudio: Integrated Development Environment for R," 2025, *Posit Software, PBC, Boston, MA*. [Online]. Available: http://www.posit.co/

[24] ITU-R, "Methods for selecting and describing attributes and terms, in the preparation of subjective tests BS Series Broadcasting service (sound)," Geneva, Mar. 2017. [Online]. Available: http://www.itu.int/ITU-R/go/patents/en

[25] L. Cai, H. Gao, and S. Ji, "Multi-Stage Variational Auto-Encoders for Coarse-to-Fine Image Generation," May 2017, [Online]. Available: http://arxiv.org/abs/1705.07202

APPENDIX

A.1 SUBJECTIVE ASSESSMENT: GENRE ACCURACY RESULTS

When we assessed how well the generated tracks matched their intended genre, Udio (63.16%) and EasyMusic (55.26%) were the only model to surpass a 50% accuracy rate. Suno is the next one in this facet (47.89%), Aiva the fourth (42.11%), and Soundraw the last (40.00%).

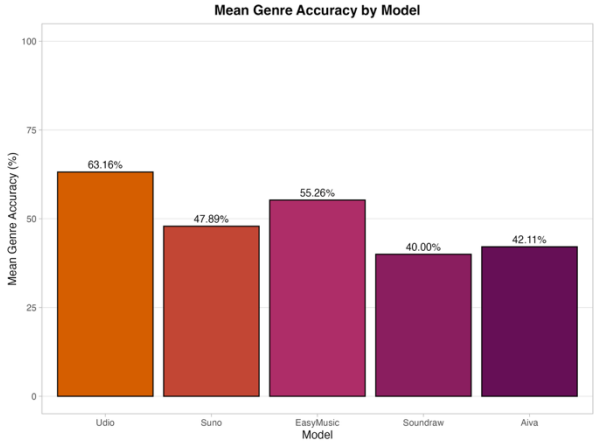

Fig. 9 Mean genre accuracy for all models

A.2 MODEL PERFORMANCE ACCROSS GENRES

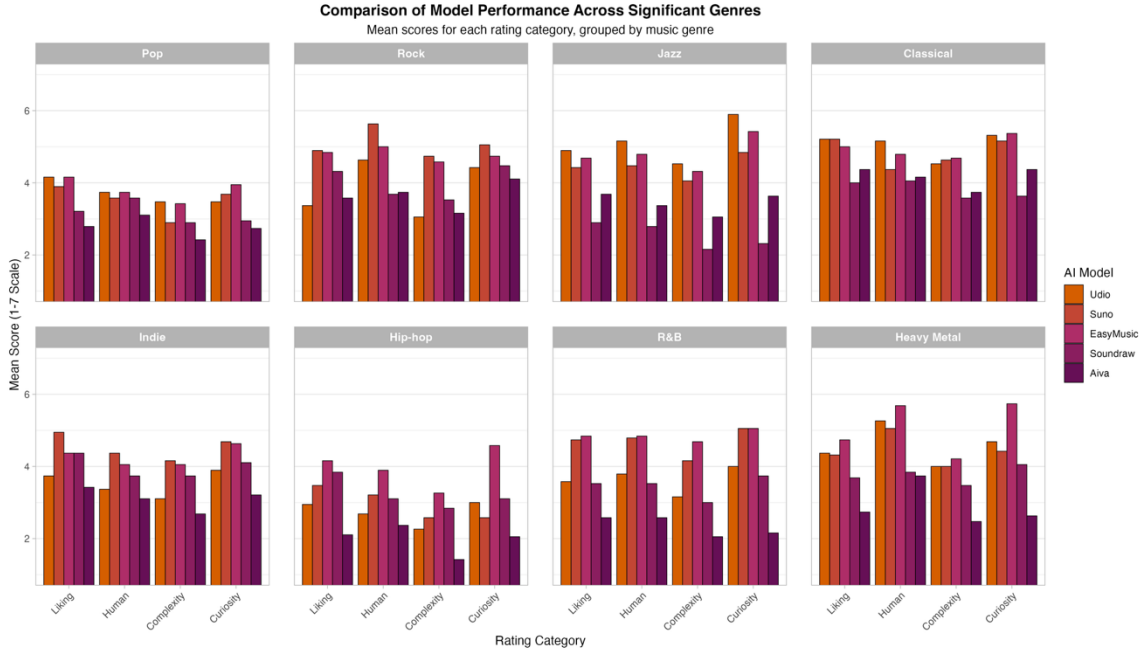

Fig. 10 Average scores for liking, perceived humanity (human), complexity and curiosity facets for all five models accross genres showing significant difference between all five.