ⵜⴰⵖⵡⴰⵏⵜ ⵏ ⵜⵙⴰⵡⴰⵍⵜ
FACULTÉ DES SCIENCES

MARRAKECH
جامعة القاضي عياض
UNIVERSITÉ CADI AYYAD

Semlalia

كليـــــــــة العلوم
FACULTY OF SCIENCE

AI-Powered Music Production: A Lyrics-to-Song Framework

Project Report

Prepared by:
Mustapha Mansouri
Abderahman El Hamidy

Supervised by:
Pr. Mohamed Amine Chadi

December 21, 2025

Contents

1 Introduction

In the rapidly evolving landscape of Artificial Intelligence, the frontier of creative synthesis has shifted from static text and imagery to the dynamic and computationally demanding domain of high-fidelity audio production. While Large Language Models (LLMs) have mastered the nuances of human syntax, the generation of music—specifically the synthesis of a cohesive "song" that combines lyrics, emotive vocals, and harmonized instrumentation—remains a significant challenge. The objective of this project is to bridge the fundamental gap between creative writing and musical realization by developing an end-to-end, GPU-accelerated pipeline capable of transforming raw text lyrics into a fully produced audio track.

Traditional music production is a multidisciplinary craft that historically requires significant human capital, including vocal performance expertise, complex instrumental arrangement, and professional mixing engineering. For many independent creators, these requirements represent a prohibitive barrier to entry. Our framework democratizes this process, simplifying the creative workflow by leveraging state-of-the-art transformer-based architectures to automate these complex tasks.

By utilizing **Suno Bark** [4] for non-linear, emotive vocal synthesis and **Meta's MusicGen** [5] for sophisticated, audio-conditioned instrumental composition, we move beyond simple text-to-audio. Our approach focuses on "Cross-Model Conditioning," a methodology that ensures the generated backing track is not merely a generic loop, but a structured accompaniment that respects the unique rhythm, pitch variance, and emotional tone of the vocals. This report details the technical architecture, the sequential integration methodology, and the final user-facing application designed to make AI-powered music production accessible to creators of all technical levels.

2 State of the Art

The field of AI-driven music generation has seen a dramatic transition from rule-based systems to deep generative models. Understanding the current state of the art is essential to situate our specific pipeline within the broader research landscape.

2.1 Evolution of Music Synthesis

Early attempts at computer-generated music focused primarily on symbolic representation, such as MIDI generation, which required external synthesizers for actual sound. The emergence of **WaveNet** [1] and subsequently **GAN-based** models marked a shift toward raw audio synthesis. However, these models often struggled with long-range temporal coherence—the ability to keep a consistent melody over more than a few seconds.

2.2 Transformer-Based Audio Modeling

The current gold standard in the industry involves the use of Large Language Model (LLM) architectures adapted for audio. Systems like **Google's MusicLM** [3] and **OpenAI's Jukebox** [2] demonstrated that treating audio as a sequence of discrete tokens (similar to words in a sentence) allows models to learn complex hierarchical structures, from high-level genre patterns down to individual instrument timbres.

2.3 Text-to-Vocal and Vocal Conditioning

In the specific domain of lyrics-to-song synthesis, the state of the art is currently defined by two major approaches:

- **Autoregressive Modeling (Suno Bark [4]):** Bark represents a breakthrough in non-linear synthesis. Unlike traditional vocoders, it generates audio tokens that capture non-verbal communication, such as laughter, sighs, and specifically, melodic singing cadence.
- **Diffusion and Latent Modeling (Meta MusicGen [5]):** MusicGen, part of the AudioCraft suite, utilizes a single-stage EnCodec model for compression and a transformer for generation. It currently leads the field in "Melody Conditioning"—the ability to take an external audio source and generate an accompaniment that respects its rhythmic and melodic constraints.

Our project utilizes these state-of-the-art architectures in a sequential pipeline, addressing a common industry gap: the lack of a unified, user-friendly framework that synchronizes specialized vocal generators with advanced instrumental composers.

3 System Architecture

The system architecture is founded on a modular, sequential dependency model designed to solve the problem of "temporal misalignment"—a common failure in AI music where vocals and instruments drift apart. Unlike "naive" systems that generate audio components in parallel, our architecture establishes the vocal track as the primary "Temporal Anchor" for the entire composition.

The pipeline is divided into two primary technical branches, as illustrated in Figure 1:

1. **The Vocal Branch (Melodic Generation):** This branch leverages the Suno Bark model, a GPT-style transformer that treats audio as a series of discrete tokens rather than traditional waveforms. By injecting specific semantic tokens—such as musical notes (♪) and [music] tags—into the input lyrics, we guide the model to interpret the text as a melodic prompt. The result is a "sung" vocal track that captures human-like nuances which serve as the blueprint for the song's tempo.
2. **The Instrumental Branch (Harmonic Alignment):** The second stage utilizes Meta's MusicGen. The critical innovation here is the use of **Melody Conditioning**. In addition to a text description defining the genre, the model receives the raw audio output from the Vocal Branch. Using a melody-matching tensor, MusicGen analyzes the vocal frequencies to generate a backing track that is tonally and rhythmically synchronized with the singer.
3. **Orchestration Layer:** A final "Composer" layer manages the mixing of these two stems. This stage handles volume normalization, stereo widening, and final file encoding, ensuring that the disparate outputs of the two neural networks are merged into a singular, high-fidelity audio experience.

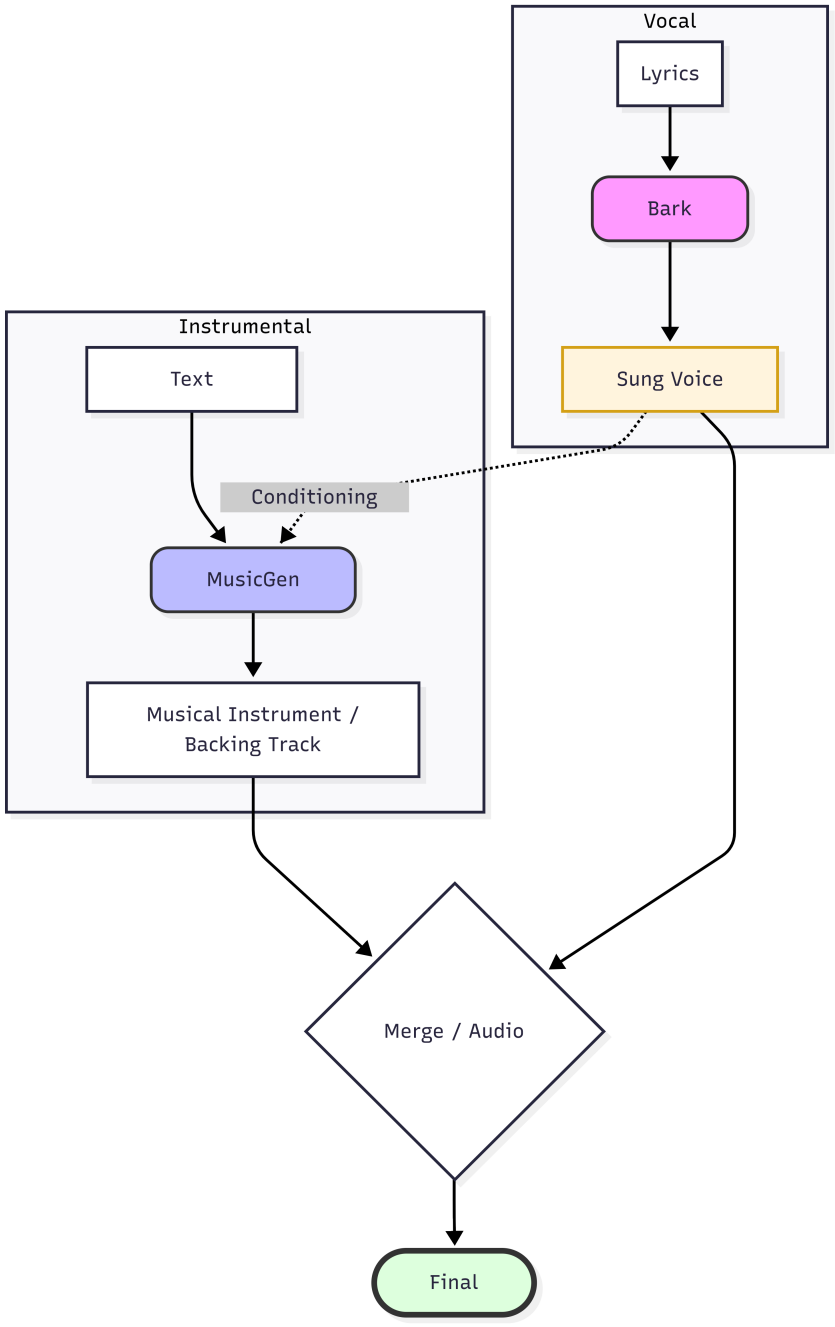

Figure 1: Workflow illustrating the dependency of the Instrumental Branch on the Vocal Branch to ensure rhythmic synchronization.

4 Methodology

The development of the "Lyrics-to-Song" pipeline was executed through three specialized Jupyter notebooks, each addressing a distinct layer of the production process. This modular approach allowed for isolated testing of vocal emotion, instrumental harmony, and final acoustic balancing.

4.1 Vocal Synthesis (Notebook 1: Bark Generation)

The first phase, implemented in `generating-sung-voice-with-Bark.ipynb`, focused on converting raw text into a melodic vocal performance. We utilized the Suno Bark model, specifically choosing to install it in a "GPU-safe" mode to preserve Kaggle's pre-configured PyTorch environment.

The core of our work here involved **Semantic Feature Engineering**. We discovered that Bark's stochastic nature requires precise prompting to trigger a singing voice rather than speech. By wrapping lyrics in musical meta-tags such as `[music]` and utilizing the eighth-note symbol (♪), we successfully biased the model's transformer layers toward melodic output. We also implemented a selection logic for **History Prompts**, identifying specific speaker presets that maintain pitch stability across longer phrases.

4.2 Instrumental Composition (Notebook 2: MusicGen)

The second phase, detailed in `instrumental-generation_2.ipynb`, utilized the Meta AudioCraft library. The primary technical hurdle was ensuring the backing track followed the unique rhythm of the AI vocals generated in the previous step.

Instead of generating a random track, we employed **Melody Conditioning**. This process involves feeding the Bark-generated `.wav` file into MusicGen's conditioning tensor. The model performs a latent analysis of the vocal track's rhythmic envelope and pitch distribution, generating an accompaniment that matches the tempo (BPM) and harmonic key of the singer. This notebook also handled style mapping, where high-level genre prompts were translated into complex acoustic descriptions for the model.

4.3 Final Orchestration

The final integration was to:

- **Peak Normalization:** Ensuring both tracks reach a consistent decibel level to prevent clipping.
- **Loudness Balancing:** Using **pydub** and **scipy** to ensure the instrumental track provides a rich bed without drowning out the vocal nuances.
- **Sample Rate Standardization:** Harmonizing the disparate outputs (24kHz from Bark and 32kHz from MusicGen) into a single high-fidelity stereo export.

5 Results and Discussion

The primary result of this project is the successful development of a comprehensive, end-to-end song generation pipeline. This system effectively automates the entire creative

process, transitioning from raw text lyrics to a finalized, mixed audio track. By integrating specialized models for both vocal and instrumental synthesis, the pipeline functions as a unified digital studio capable of producing complete musical works from simple text inputs.

5.1 Application Interface

To bridge the gap between our technical notebooks and the end-user, we developed a streamlined interface. As seen in Figure 2, the interface focuses on "Creative Abstraction," where the user only needs to provide the lyrical content and a genre preference.

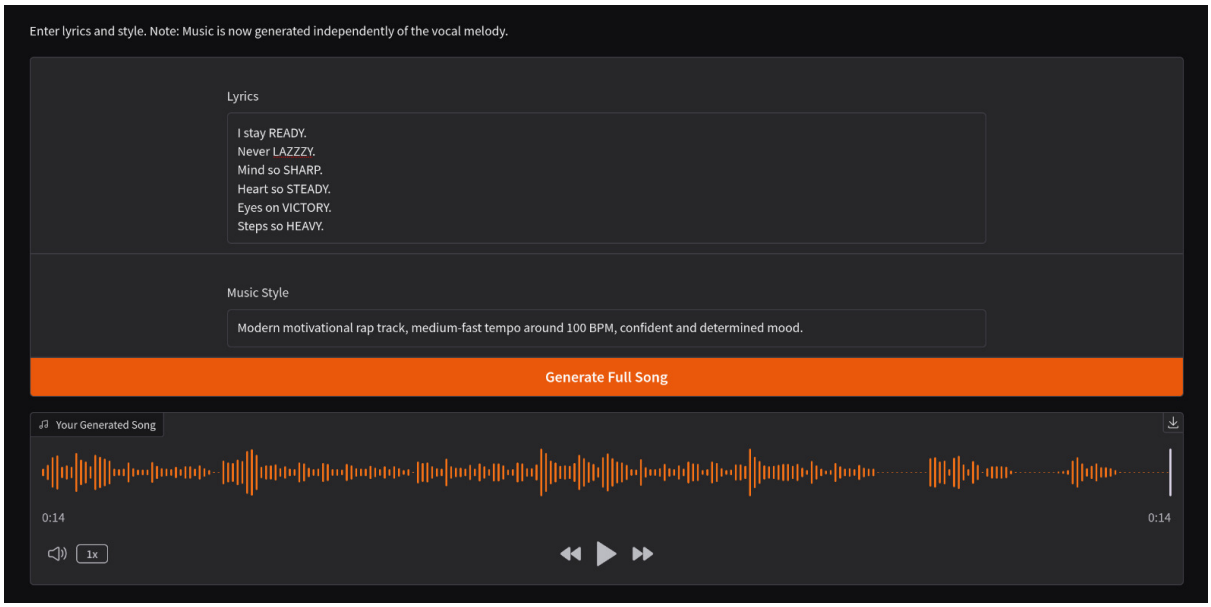

Figure 2: The integrated web interface showing the lyric input area, style selection, and generation status.

The application successfully hides the complexity of tokenization and conditioning, providing an "Input-to-Audio" experience. The inclusion of an integrated player allows for immediate qualitative feedback, which is essential for iterative songwriting.

5.2 Discussion of Audio Quality and Alignment

Qualitatively, the outputs exhibit a "studio-like" atmosphere. The MusicGen backing tracks are remarkably successful at masking the digital artifacts sometimes present in AI singing, creating a cohesive acoustic experience.

However, a critical observation during testing was the presence of **Alignment Drift**. While the melody-conditioning variant of MusicGen is powerful, it occasionally fails to sync perfectly with the lyrical phrasing of the Bark model in complex or fast-paced song structures. Sometimes the instrumental swell does not perfectly align with the vocal climax, leading to a slight rhythmic disconnect. This suggests that while the models are "aware" of each other through conditioning, they do not yet share a unified temporal clock.

6 Conclusion

This project successfully demonstrates that high-quality, end-to-end music production is achievable through the intelligent coupling of specialized transformer models. By establishing a sequential dependency between Suno Bark for emotive vocals and Meta MusicGen for melody-conditioned accompaniment, we have created a functional framework that respects the structural nuances of human songwriting.

6.1 Future Work

Our future roadmap focuses on evolving this pipeline from a proof-of-concept into a professional-grade music production tool by addressing the following areas:

- **Advanced Temporal Alignment:** Our primary objective is to eliminate the "alignment drift" identified during the results phase. We aim to implement a more robust feedback loop using beat-detection algorithms to ensure that the vocal phrasing produced by Bark is mathematically quantized to the instrumental rhythm generated by MusicGen.
- **Extended Song Duration:** To move beyond the current 30-second limit, we plan to implement a recursive "Sliding Window" generation technique. This will allow the model to generate full-length tracks (3–5 minutes) while maintaining melodic and tonal consistency by using the end of one segment as the contextual anchor for the next.
- **Multi-Track Synthesis and Mixing:** Future iterations will focus on generating individual audio stems (separate tracks for drums, bass, and melody) rather than a single stereo file. This will allow the "Composer" phase to perform more granular audio engineering, such as independent EQ, panning, and professional mastering, resulting in a much higher fidelity final output.
- **Voice Personalization:** We aim to explore the integration of fine-tuning techniques that allow users to adapt the vocal branch to specific voices or styles, significantly increasing the creative utility and personalization of the framework.

Through these improvements, we aim to transform this framework into a comprehensive tool that makes high-quality music production accessible to any creator with a story to tell.

References

[1] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] P. Dhariwal *et al.*, "Jukebox: A Generative Model for Music," *arXiv preprint arXiv:2005.00341*, 2020.

[3] A. Agostinelli *et al.*, "MusicLM: Generating Music From Text," *arXiv preprint arXiv:2301.11325*, 2023.

[4] Suno AI, "Bark: A Transformer-based Text-to-Audio Model," GitHub repository, 2023. [Online]. Available: https://github.com/suno-ai/bark.

[5] J. Copet *et al.*, "Simple and Controllable Music Generation," *arXiv preprint arXiv:2306.05245*, 2023.

[6] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," *arXiv preprint arXiv:2210.13438*, 2022.

[7] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.