**A Technical Blueprint for an Open-Source Multimodal AI Music Generation Application**

**1. Introduction: The Landscape of AI Music Generation**

The intersection of artificial intelligence and creative industries has witnessed a significant surge in recent years, with music generation standing out as a particularly dynamic field 1. The ability of AI to analyze vast datasets of music and subsequently produce novel compositions has captured the imagination of both technologists and artists 3. This growing interest stems from the potential of AI not only to democratize music creation, making it accessible to individuals without formal training, but also to augment the workflows of professional musicians, offering new avenues for experimentation and inspiration 1. While the debate continues regarding AI's capacity to truly replicate the emotional depth and nuanced creativity of human composers, its role as a powerful tool in the music creation process is becoming increasingly evident 1.

This report outlines a technical blueprint for the development of a comprehensive open-source music generation application. This application aims to harness the latest advancements in machine learning and multimodal AI to provide a versatile platform for music creation, catering to a wide range of user needs and technical expertise. The core requirements for this application, as specified in the user's query, include the integration of various machine learning techniques, the incorporation of multimodal AI approaches, and the provision of a single user interface for both inference (music generation) and training (model fine-tuning) [User Query]. Furthermore, the application should facilitate seamless data ingestion from various sources, enable fine-tuning of existing models based on user interaction and new data, and leverage cutting-edge technologies. Recognizing the constraints of local development on a Mac Mini M4 (16GB), the architecture will be designed to offload computationally intensive training tasks to Google Colab, thereby balancing resource limitations with the need for powerful processing capabilities [User Query]. The open-source nature of this project is paramount, fostering community collaboration and driving innovation within the field of AI music generation 6. The application, therefore, is envisioned as a cutting-edge, community-driven platform that empowers users to explore the vast possibilities of AI in music, utilizing the strengths of both local and cloud computing environments.

**2. Foundational Machine Learning Techniques for Music Generation**

The creation of music through artificial intelligence relies on a variety of machine learning techniques, each with its own strengths and characteristics. Autoregressive models, diffusion models, variational autoencoders, and generative adversarial networks represent some of the foundational approaches in this domain.

**2.1 Autoregressive Models (RNNs, Transformers):** Autoregressive models are designed to predict the next element in a sequence based on the preceding elements, making them inherently well-suited for the sequential nature of music 7. In the context of music generation, these models learn to predict the next note, chord, or audio sample given the history of the composition. Recurrent Neural Networks (RNNs) are a class of neural networks that excel at processing sequential data by maintaining an internal state that captures information about past inputs 4. Long Short-Term Memory (LSTM) networks, a specialized type of RNN, address the challenge of capturing long-range dependencies in sequences, allowing them to learn musical structures that unfold over extended periods 7.

More recently, Transformer models have gained prominence in music generation due to their ability to capture both short-term and long-term dependencies with remarkable effectiveness, and their inherent parallelism allows for efficient training and inference 7. Transformer architectures utilize attention mechanisms that allow the model to weigh the importance of different parts of the input sequence when making predictions 8. MusicGen, a state-of-the-art autoregressive Transformer model developed by Meta AI, has demonstrated exceptional capabilities in generating high-quality music conditioned on text descriptions or melodic features 6. Its architecture is notable for its single-stage design and efficient token interleaving, which streamlines the generation process 12. MusicGen supports both text-to-music generation, where users can provide textual prompts to guide the style and content of the music, and melody conditioning, where an existing melody can be used as a basis for generating new music 11. Another example of a Transformer-based approach is TchAIkovsky, a model specifically designed for generating MIDI files of piano music, showcasing the versatility of Transformers in handling symbolic music representations 14. The ability of Transformer models, particularly exemplified by MusicGen, to produce high-quality music and effectively handle conditional inputs makes them a strong foundation for the proposed application.

**2.2 Diffusion Models:** Diffusion models offer a different paradigm for music generation. These models work by gradually adding noise to a musical data sample until it becomes pure noise, and then training a neural network to reverse this process, learning to denoise the noisy data back into a coherent musical piece 8. Stable Audio, developed by Stability AI, is an example of a diffusion model that has shown promise in generating audio content 8. While the commercial version of Stable Audio can produce full musical tracks, Stable Audio Open, its open-source counterpart, specializes in generating shorter audio samples, sound effects, and production elements, making it a valuable tool for creating specific sonic textures 15. The open-source nature of Stable Audio Open also makes it a candidate for fine-tuning on custom datasets. Jukebox Diffusion represents another notable approach, utilizing a hierarchical latent diffusion model to generate longer and more complex musical passages 17. Diffusion models like Stable Audio offer a compelling alternative for generating diverse audio content, particularly sound effects and shorter musical segments, which could complement the capabilities of autoregressive models within the application.

**2.3 Variational Autoencoders (VAEs):** Variational Autoencoders (VAEs) are generative models that learn a compressed representation of the input data in a latent space 7. This latent space is structured in a way that allows for meaningful interpolation and sampling, enabling the generation of variations of existing musical pieces or the creation of new music by sampling from the learned latent distribution 8. MusicVAE, developed by Google Magenta, is an example of an LSTM-based VAE that has been successfully applied to music generation, demonstrating the ability to learn latent representations of musical sequences 7. VAEs can be particularly valuable for tasks such as generating variations on a musical theme or creating smooth transitions between different musical styles or sections within a composition.

**2.4 Generative Adversarial Networks (GANs):** Generative Adversarial Networks (GANs) consist of two neural networks: a generator and a discriminator 9. The generator's task is to create new data samples, in this case, music, while the discriminator's task is to distinguish between the generated samples and real music from the training data. These two networks are trained in an adversarial manner, with the generator trying to fool the discriminator, leading to the generation of increasingly realistic and high-quality musical outputs 10. GANs have shown effectiveness in capturing intricate details and generating realistic outputs in various domains, including image and audio synthesis 19. While GANs could be explored for generating highly realistic audio within the application, their training process can often be more complex and less stable compared to other generative models.

**3. Leveraging Multimodal AI for Enhanced Music Creation**

To create a truly comprehensive and versatile music generation application, it is essential to leverage the power of multimodal AI, allowing users to interact with the system using various forms of input, including text, audio, MIDI, and potentially even images and videos.

**3.1 Incorporating Text as a Modality:** Text-to-music generation is a prominent area within AI music, enabling the translation of semantic meaning expressed in natural language into musical compositions 6. This capability allows users to describe the desired music in terms of genre, mood, instrumentation, tempo, and other descriptive parameters, and have the AI generate music that aligns with these specifications 8. Models like MusicGen 8, MusicLM 8, and Stable Audio 15 have demonstrated significant progress in this area, allowing users to provide textual prompts and receive corresponding musical outputs. For instance, MusicGen excels at generating high-quality music from text prompts, offering control over various musical attributes 12. The effectiveness of text-to-music generation often relies on the quality and specificity of the text prompts provided by the user 19. Techniques like prompt engineering, which involves crafting detailed and well-structured prompts, can significantly influence the generated music, allowing users to achieve more precise and desired results 24. Integrating text as an input modality provides a versatile and intuitive way for users to guide the music generation process, regardless of their musical background or technical expertise.

**3.2 Incorporating Audio as a Modality:** Audio input offers another powerful way to guide AI music generation. Audio-to-audio generation techniques enable tasks such as style transfer, where the stylistic characteristics of one piece of music are applied to another 25, music continuation, where an existing musical idea is extended by the AI 25, and remixing, where elements of an audio track are manipulated or combined in new ways 17. MusicGen further supports melody conditioning through audio prompts, allowing users to provide an existing melody, either sung, hummed, or played, as a basis for generating a new musical arrangement 11. Tools like BassNet utilize AI to generate basslines that complement an uploaded audio track, demonstrating the potential for creating specific instrumental parts from audio input 27. The use of audio for style transfer is a particularly active area of research, with various techniques being explored, including those based on CycleGANs 28, Latent Diffusion Models (LDMs) 29, and MIDI-based approaches like Groove2Groove 30. Notably, training-free style transfer methods using pre-trained LDMs have emerged, offering the ability to transfer musical style without the need for extensive fine-tuning 29. Incorporating audio input allows users, especially musicians, to interact with the application using their existing musical ideas, facilitating style imitation, continuation, and the creation of specific instrumental elements in a more direct and intuitive manner.

**3.3 Integrating MIDI as a Modality:** MIDI (Musical Instrument Digital Interface) is a widely used symbolic representation of music that encodes information about notes, timing, and musical parameters 31. Its structured and editable format makes it highly suitable for both training AI music models and for generating music with precise control over musical elements 9. Models like MuseNet 7 and Music Transformer 7 are examples of AI systems that can generate music in MIDI format, learning from large datasets of MIDI files. Furthermore, a variety of AI MIDI generators and plugins are available, offering functionalities such as melody and chord progression generation 25. The Groove2Groove system specifically focuses on MIDI style transfer, enabling the application of the stylistic characteristics of one MIDI file to another 30. Research also explores MIDI conditioned music generation, where MIDI input can be used to guide the generation of new musical sequences 34. Integrating MIDI as an input modality provides users with a structured and editable format for interacting with the AI, allowing for precise control over musical parameters and enabling tasks like style transfer and harmonization with greater accuracy.

**3.4 Exploring Other Potential Modalities (Images, Videos):** The field of multimodal AI extends beyond text and audio, encompassing other modalities like images and videos, which can also serve as powerful sources of inspiration for music generation. MuMu-LLaMA is a notable model that demonstrates the potential for multimodal music understanding and generation from text, images, and videos, showcasing the ability to connect diverse sensory inputs with musical outputs 6. Image-to-music generation, where musical pieces are created based on the content or style of an image, is an emerging area 25. Similarly, video-to-music generation explores the creation of soundtracks or musical pieces that are contextually relevant to the visual content of a video 6. While these modalities are still under active research and development in the context of music generation, their incorporation into the application could unlock entirely new creative possibilities, allowing users to draw inspiration from visual sources and create music that is synesthetically linked to images or videos.

**4. Designing a Unified Application Interface for Inference and Training**

A key aspect of this project is the development of a unified application interface that seamlessly integrates functionalities for both music generation (inference) and model fine-tuning (training). This single interface should provide a user-friendly experience for interacting with the AI models and managing the entire music creation process.

**4.1 User-Friendly Interface for Model Interaction:** The application must feature an intuitive and accessible design that allows users to easily input various types of prompts, including text, audio, MIDI, image, and video, depending on the chosen model and desired outcome 9. Clear controls should be provided for adjusting generation parameters such as genre, tempo, instrumentation, key, and mood 25. Visual feedback, such as waveform visualizations for audio and notation displays for MIDI, can enhance the user experience 38. Real-time adjustments of parameters, where changes are immediately reflected in the generated music, can further facilitate experimentation and creative exploration 37. The interface should be designed to abstract away the underlying complexities of the AI models, presenting users with a streamlined and enjoyable pathway to music creation, whether they are generating new pieces or fine-tuning existing models.

**4.2 Integrated Training Management:** The application should provide a comprehensive workflow for managing the entire model fine-tuning process. This includes intuitive tools for data ingestion from various sources, preprocessing functionalities to prepare the data for training, and clear steps for organizing the data into training, validation, and testing sets 6. Users should have the option to select from a range of pre-trained models, such as MusicGen and Stable Audio, for fine-tuning on their own datasets 11. The interface should allow for the adjustment of key training parameters, including the number of epochs, batch size, and learning rate 20. Given the computational demands of training large music generation models, seamless integration with Google Colab for offloading these heavy tasks is crucial 26. The application should display the training progress in real-time, allowing users to monitor the learning process 40. Functionalities for saving training checkpoints and resuming interrupted training sessions are also essential for a robust training management system 40. By streamlining the fine-tuning process from data preparation to model training and monitoring, the application can make this powerful technique accessible even to users who may have limited prior experience with machine learning.

**4.3 Interface for Inference and Output Management:** The inference section of the application should allow users to easily select either pre-trained models or models they have fine-tuned for music generation. Clear controls for setting inference parameters, such as the desired length of the generated music and the sampling strategy to be used, should be provided [User Query]. The generated music should be displayed in a user-friendly manner, offering options for both audio visualization (e.g., waveform) and symbolic representation (e.g., MIDI notation), where applicable [User Query]. Users should be able to readily play back the generated music and download the output in various formats, including audio files (e.g., WAV, MP3) and MIDI files 36. Efficient management of generated outputs, including options for saving, organizing, and tagging musical pieces, will further enhance the user experience. The inference interface should be designed to be straightforward and efficient, enabling users to quickly generate music and manage their creative output in a variety of formats.

**5. Project Architecture for Local Development and Cloud Training**

To effectively develop and deploy the music generation application, a well-defined project architecture is necessary, taking into account the constraints of local development on a Mac Mini M4 and the need to leverage cloud resources for heavy training.

**5.1 Local Development on Mac Mini M4 (16GB):** Development on the Mac Mini M4 with 16GB of RAM necessitates a focus on efficiency and resource optimization. For local inference tasks, the application should prioritize the use of lightweight AI models and efficient inference techniques that can run smoothly on consumer-grade hardware 42. Frameworks and libraries optimized for Apple's silicon, such as PyTorch with the Metal backend, should be considered to maximize performance. Efficient data loading and preprocessing pipelines will also be crucial for ensuring a responsive user experience during local development. The local development environment should primarily be used for rapid prototyping of the user interface and user experience, as well as for testing inference with smaller models or shorter music generation lengths.

**5.2 Offloading Heavy Training to Google Colab:** Recognizing the computational demands of training large-scale music generation models, the application's architecture should facilitate a seamless workflow for offloading these tasks to Google Colab [User Query]. This will require a system for easily transferring training data from the local machine to Google Colab's environment. Google Colab's access to powerful GPU resources will significantly accelerate the training process for larger models and datasets 40. The architecture must also include mechanisms for synchronizing the trained model weights back from Google Colab to the local machine, allowing the locally developed application to utilize the full potential of the cloud-trained models for inference. This hybrid approach leverages the strengths of both local development for UI/UX and cloud computing for intensive model training.

**5.3 Modular and Extensible Design:** To ensure the long-term maintainability and scalability of the open-source project, a modular architecture is essential. The application should be designed with distinct, independent modules for different functionalities, such as data ingestion, preprocessing, model selection, training, inference, and user interface. This modularity will allow for the easy integration of new machine learning models, support for additional input modalities, and the addition of new features over time. Well-defined Application Programming Interfaces (APIs) and interfaces between these modules will promote flexibility and facilitate contributions from the open-source community.

**5.4 Technology Stack:** The choice of the technology stack will significantly impact the project's development and performance. Python is a natural choice as the primary programming language due to its extensive ecosystem of machine learning libraries and tools. For the implementation of machine learning models, frameworks like PyTorch and TensorFlow are leading options, offering robust support for training and inference. The user interface can be built using frameworks such as Gradio or Streamlit, which allow for the rapid creation of interactive web applications suitable for both inference and training management. For audio-specific tasks, libraries like Librosa can be used for feature extraction and processing 44, while Music21 provides powerful tools for working with MIDI data 46. PyDub can be utilized for various audio file manipulations 48. The selection of these technologies considers factors such as performance, ease of use, a large and active community for support, and their relevance to the domain of AI music generation.

**6. Data Ingestion and Management Strategies**

Effective data management is crucial for training and utilizing machine learning models for music generation. The application needs to support the ingestion of various music data formats, provide tools for preprocessing, facilitate dataset creation, and potentially leverage existing music datasets.

**6.1 Supporting Various Music Data Formats:** To cater to a wide range of users and data sources, the application should support common audio file formats such as MP3, WAV, and FLAC 9. Support for MIDI files is also essential, given their prevalence in symbolic music representation 31. Furthermore, as the application aims to incorporate multimodal AI, it should be designed to handle other potential modalities like image and video formats, even if initial support is limited 6. This broad format support will allow users to easily bring their existing music libraries and creative content into the application.

**6.2 Data Preprocessing Techniques:** Before music data can be effectively used for training AI models, it often requires preprocessing. For audio data, this may include steps such as resampling audio files to a consistent sample rate, normalizing the audio levels, trimming silence from the beginning and end of tracks, and segmenting longer audio files into shorter, manageable chunks 33. For MIDI data, preprocessing might involve quantization of note timings or pitch shifting to augment the dataset. The application should also consider incorporating techniques for handling noisy or low-quality audio data, potentially through noise reduction algorithms 19. Providing users with a suite of preprocessing tools will ensure that their data is in an optimal format for training and generation.

**6.3 Dataset Creation and Management:** The application should guide users through the process of organizing their music data into datasets suitable for machine learning. This includes the ability to split the data into distinct sets for training the model, validating its performance during training, and testing its final capabilities on unseen data 20. Support for data augmentation techniques, such as adding slight variations to the audio or MIDI data, can help to increase the size and diversity of the training dataset, which can improve the model's generalization ability 33. For users working with very large datasets, the application should consider options for integrating with cloud storage solutions like Google Cloud Storage to efficiently manage and access the data.

**6.4 Leveraging Existing Music Datasets:** To reduce the burden of collecting and preparing large amounts of training data from scratch, the application should provide options for users to integrate with publicly available music datasets 31. Datasets like the MAESTRO dataset, which contains a large collection of piano performances in MIDI format 31, and the Lakh MIDI Dataset, a more extensive collection of MIDI files across various genres 32, can be valuable resources for pre-training models or for fine-tuning existing models on specific musical styles. Providing easy access to and integration with such datasets can significantly accelerate the development and personalization of music generation models within the application.

**7. Fine-tuning Pre-trained Models for Personalized Music Generation**

A key feature of the application is the ability for users to fine-tune pre-trained music generation models on their own data. This allows for the creation of personalized models that can generate music in specific styles or based on individual preferences.

**7.1 Fine-tuning Existing Models (MusicGen, Stable Audio):** The application should provide clear and intuitive workflows for fine-tuning popular open-source music generation models like MusicGen and Stable Audio 39. This includes user-friendly interfaces for selecting the pre-trained model to be fine-tuned, uploading the user's custom music dataset, and adjusting the key parameters that control the fine-tuning process 11. The application should offer options for different fine-tuning strategies, such as full fine-tuning, where all the model's parameters are updated, and Low-Rank Adaptation (LoRA), a more parameter-efficient technique that can lead to faster training and smaller fine-tuned models 41. Users should have the ability to control important training parameters like the learning rate, which determines the step size during optimization, and the number of training epochs, which dictates how many times the model iterates over the training data 20. By providing a simplified and well-documented fine-tuning process, the application can empower users to create highly personalized music generation systems tailored to their specific needs and artistic visions.

**7.2 Adapting Models Based on Usage and User Feedback:** To further enhance the personalization of the music generation experience, the application should incorporate mechanisms for adapting the models based on user usage and feedback. This could involve collecting explicit feedback, such as user ratings or selections of generated music, or implicitly tracking user preferences based on their interaction history with the application. Techniques such as reinforcement learning, where the model is trained to maximize a reward signal derived from user feedback, or active learning, where the model strategically selects the most informative data points for further training, could be explored. By continuously learning from user interactions, the application can refine its models over time to better align with individual user tastes and creative goals.

**7.3 Fine-tuning with New Incoming Data:** The ability for users to continuously train their personalized music generation models with new incoming data is crucial for maintaining the relevance and evolving the style of the generated music. The application should provide a straightforward system for users to add new music data to their existing fine-tuned models 41. Implementing incremental fine-tuning strategies, which allow the model to be updated with new data without requiring a complete retraining from scratch, will be important for efficiency 40. This continuous learning capability will ensure that the generated music can adapt to the user's changing preferences and incorporate new musical influences over time.

**8. Incorporating Latest Cutting-Edge Technologies**

To remain at the forefront of AI music generation, the application should strive to incorporate the latest cutting-edge technologies and models emerging in the field.

**8.1 Exploring Recent Advances in AI Music Generation:** Several recent advancements in AI music generation offer exciting possibilities for the application. DiffRhythm, for example, is a model that has demonstrated the ability to generate full-length songs with impressive speed 52. YuE represents another significant development, focusing on the challenging task of transforming lyrics into complete songs, including both vocal and accompaniment tracks 24. VMB (Visual Music Bridge) is a framework that utilizes explicit bridges of text and music for multimodal alignment, showing promising results in generating music from diverse inputs like images and videos 35. Exploring the integration of such advanced models can significantly enhance the application's capabilities and provide users with access to state-of-the-art music generation techniques.

**8.2 Investigating Open-Source AI Singing Voice Synthesis Models:** The ability to generate realistic singing voices is a highly sought-after feature in music generation. Several open-source AI singing voice synthesis models are becoming increasingly sophisticated. Models like XTTS, ChatTTS, MeloTTS, and OpenVoice offer capabilities for generating speech and, in some cases, singing voices with varying degrees of naturalness and multilingual support 54. Tools like ACE Studio and Synthesizer V provide more specialized solutions for AI singing voice synthesis, often with granular control over vocal parameters 55. Musicfy is another platform that focuses on AI voice generation and even allows for voice cloning 57. Investigating and potentially integrating one or more of these open-source singing voice synthesis models could significantly enhance the application by enabling the generation of complete songs with vocals, opening up new creative avenues for users.

**8.3 Utilizing Audio-to-MIDI Conversion Tools:** Audio-to-MIDI conversion tools can play a valuable role in a comprehensive music generation application. These tools can convert audio recordings of melodies or instrumental parts into MIDI format, which can then be used as input for MIDI-based generation models or for further manipulation within the application 59. Basic Pitch 59 and Samplab 59 are examples of open-source and readily available audio-to-MIDI conversion tools that could be integrated. This functionality would allow users to easily capture musical ideas by singing or playing an instrument and then use these recordings to guide the AI music generation process.

**8.4 Exploring Stem Separation Techniques:** Stem separation techniques, which involve separating a mixed audio recording into its individual instrument or vocal tracks (stems), can be beneficial for various aspects of the application 61. Tools like LANDR Stems 61, Lalal.ai 25, and Music AI 62 utilize AI algorithms to perform this separation. Stem separation could be used during data preprocessing to isolate instrumental tracks for training models focused on specific instruments. It could also enable features like remixing, where users can isolate and manipulate individual parts of a song. Integrating stem separation capabilities would add another layer of functionality and flexibility to the application.

**9. Open-Source Implementation Considerations**

Given the goal of creating an open-source music generation application, several implementation considerations related to licensing, community engagement, and documentation are paramount for the project's success.

**9.1 Licensing and Distribution:** Choosing an appropriate open-source license is a critical first step. Licenses like the Apache 2.0 license or the MIT license offer a balance between freedom of use and requirements for attribution 53. The project's licensing should be carefully considered to encourage both usage and contributions from the community. A clear plan for distributing the application and any trained model weights is also necessary. Platforms like GitHub for code hosting and the Hugging Face Hub for sharing models are standard choices within the open-source AI community 51. Utilizing these platforms will facilitate accessibility and collaboration.

**9.2 Community Engagement and Contributions:** A thriving open-source project relies heavily on the engagement and contributions of its community. Establishing clear guidelines for how individuals can contribute to the project, whether through code submissions, data contributions, documentation improvements, or providing feedback, is essential. Fostering a welcoming and collaborative environment through forums, issue tracking systems, and a well-defined pull request process will encourage community involvement and help drive the project forward.

**9.3 Documentation and Tutorials:** Comprehensive and well-maintained documentation is crucial for any open-source project. This includes clear instructions for installation, detailed explanations of the application's features and functionalities, and guidance for developers who wish to contribute to the codebase. Additionally, creating tutorials and practical examples that walk users through various music generation tasks will help them get started quickly and effectively 40. High-quality documentation and readily available learning resources will significantly enhance the usability and adoption of the application within the broader community.

**10. Conclusion and Future Directions**

The development of a comprehensive open-source multimodal AI music generation application presents a significant yet exciting undertaking. By leveraging the power of autoregressive models like MusicGen and exploring the potential of diffusion models and other generative techniques, the application can provide a robust foundation for music creation. The integration of multiple input modalities, including text, audio, MIDI, and potentially images and videos, will offer users unparalleled flexibility and creative control. A unified and user-friendly interface, coupled with seamless training management capabilities through Google Colab, will make the power of AI music generation accessible to a wide audience. The project's open-source nature, underpinned by careful consideration of licensing, community engagement, and thorough documentation, will foster collaboration and drive innovation in this rapidly evolving field.

Looking ahead, several potential future enhancements and research directions could further enrich the application. Exploring advanced techniques for controlling the creativity and musicality of the generated output, such as incorporating parameters for emotional expression or structural complexity 1, would be valuable. Investigating methods for enabling the generation of music with specific artistic influences, allowing users to emulate the styles of their favorite musicians 2, could add another layer of personalization. Further research into personalized music generation based on individual user preferences and listening habits 3 could lead to highly tailored musical experiences. Addressing the challenge of generating longer and more coherent musical compositions 64 remains an important area for future development. Finally, as AI music generation continues to advance, careful consideration of the ethical implications related to copyright, originality, and the potential cultural impact of this technology 1 will be crucial for responsible innovation.

**Table 1: Comparison of Key Machine Learning Models for Music Generation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Type** | **Key Characteristics** | **Strengths** | **Weaknesses** | **Suitable Use Cases in the Application** | **Relevant Snippets** |
| Autoregressive (RNN, LSTM) | Predicts the next element in a sequence based on previous elements; RNNs handle sequential data, LSTMs capture long-term dependencies. | Good for capturing temporal structure in music; LSTMs can model longer musical phrases. | Can struggle with very long sequences; RNNs can have vanishing/exploding gradients. | Generating melodies, short musical sequences, potentially as components in larger models. | 4 |
| Autoregressive (Transformer, MusicGen) | Uses attention mechanisms to weigh different parts of the input; MusicGen is a state-of-the-art model for text/melody to music. | Excellent at capturing long-range dependencies; parallel processing; MusicGen generates high-quality, controllable music. | Can be computationally intensive for very long sequences. | Core model for text-to-music and melody-conditioned music generation. | 4 |
| Diffusion Models (Stable Audio, Jukebox Diffusion) | Generates data by reversing a noise addition process; Stable Audio for short samples/effects, Jukebox Diffusion for longer music. | Can produce high-quality and diverse audio; Stable Audio Open is open-source and fine-tunable. | Can be computationally intensive during generation. | Generating sound effects, short musical clips, potentially for style transfer or remixing. | 8 |
| Variational Autoencoders (VAEs) | Learns a latent space representation of data, enabling generation of variations and interpolations. | Good for learning underlying data structures; can generate smooth transitions and variations. | Generated samples can sometimes be less sharp or detailed than other methods. | Generating variations of existing music, style interpolation, creating latent space for musical attributes. | 7 |
| Generative Adversarial Networks (GANs) | Consists of a generator and a discriminator trained in an adversarial manner. | Can capture intricate details and generate highly realistic outputs. | Training can be unstable and require careful tuning. | Potentially for high-fidelity audio generation or style transfer tasks. | 4 |

**Table 2: Overview of Multimodal AI Approaches for Music Generation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modality** | **Description** | **Key Models/Techniques** | **Potential Benefits for the Application** | **Relevant Snippets** |
| Text | Using natural language descriptions to guide music generation. | MusicGen, MusicLM, Stable Audio, Transformer models. | Versatile and intuitive way for users to control music generation based on various attributes. | 6 |
| Audio | Using existing audio as input for style transfer, continuation, melody conditioning, or instrumental part generation. | MusicGen (melody conditioning), BassNet, Groove2Groove, Latent Diffusion Models, CycleGANs. | Allows users to guide generation using their own musical ideas, enabling style imitation and expansion. | 11 |
| MIDI | Using symbolic music representation for precise control over musical elements and enabling style transfer/harmonization. | MuseNet, Music Transformer, AI MIDI Generators, Groove2Groove. | Provides structured and editable format for interaction, facilitating precise control and specific musical tasks. | 7 |
| Image/Video | Using visual content as inspiration for music generation or for creating soundtracks. | MuMu-LLaMA, image-to-music techniques, video-to-music techniques. | Opens up new creative possibilities by linking music to visual stimuli. | 6 |

**Table 3: Comparison of Open-Source AI Singing Voice Synthesis Models**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Name** | **Key Features** | **Language Support** | **Licensing** | **Relevant Snippets** |
| XTTS | Voice cloning with minimal input (6-second audio clip), emotion and style transfer, low-latency performance. | 17 languages. | MIT. | 54 |
| ChatTTS | Designed for conversational applications, high-quality speech. | Chinese and English. | Open-source (details not specified in snippet). | 54 |
| MeloTTS | High-quality, multilingual, optimized for real-time inference on CPUs. | Wide range, including English dialects. | MIT. | 54 |
| OpenVoice v2 | Instant voice cloning from a short audio clip, flexible voice style control, zero-shot cross-lingual cloning. | Multiple languages. | MIT. | 54 |

**Table 4: List of Recommended Technology Stack Components**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Type** | **Name** | **Description** | **Justification** | **Relevant Snippets** |
| Programming Language | Python | Versatile language with a rich ecosystem for machine learning and audio processing. | Widely used in AI research and development; numerous relevant libraries available. | - |
| Framework | PyTorch | Open-source machine learning framework with strong GPU acceleration and a large research community. | Offers flexibility and performance for training and deploying AI models. | 11 |
| Framework | TensorFlow | Another popular open-source machine learning framework with robust production capabilities. | Provides scalability and a wide range of tools for building and deploying AI models. | 18 |
| UI Framework | Gradio | Python library for quickly creating interactive web interfaces for machine learning models. | Simple and efficient for building user-friendly demos and applications. | 41 |
| UI Framework | Streamlit | Another Python library for building interactive data science and machine learning applications with minimal code. | Easy to use and well-suited for creating interactive dashboards and tools. | - |
| Audio Library | Librosa | Python library for audio analysis and processing. | Provides functions for feature extraction, time-series manipulation, and more. | 44 |
| MIDI Library | Music21 | Python toolkit for computer-aided musicology, including MIDI processing. | Offers comprehensive tools for parsing, manipulating, and analyzing MIDI files. | 46 |
| Audio Library | PyDub | Python library for manipulating audio files with a simple interface. | Useful for tasks like format conversion, trimming, and volume adjustment. | 48 |

**Works cited**

1. Protecting Human Creativity in AI-Generated Music with the Introduction of an AI-Royalty Fund | GRUR International | Oxford Academic, accessed March 14, 2025, <https://academic.oup.com/grurint/article/73/12/1137/7832810>

2. AI in the Music Industry: Transforming Music Production, Discovery, and Data - DataArt, accessed March 14, 2025, <https://www.dataart.com/blog/ai-in-the-music-industry-transforming-music-production-discovery-and-data-by-sergey-bludov>

3. The Revolution of Generative AI Music: Opportunities and Challenges - MusicHub, accessed March 14, 2025, <https://www.music-hub.com/en-blog/the-revolution-of-generative-ai-music-opportunities-and-challenges>

4. The Rise of AI-Generated Music: What It Means for Artists - Flourish$Prosper, accessed March 14, 2025, <https://flourishprosper.net/music-resources/the-rise-of-ai-generated-music-what-it-means-for-artists/>

5. The Ultimate Guide to Generative AI in Music Production - Soundful, accessed March 14, 2025, <https://soundful.com/ultimate-guide-to-generative-ai-music-production/>

6. MuMu-LLaMA: Multi-modal Music Understanding and Generation via Large Language Models - arXiv, accessed March 14, 2025, <https://arxiv.org/html/2412.06660v1>

7. Intelligent Text-Conditioned Music Generation - arXiv, accessed March 14, 2025, <https://arxiv.org/html/2406.00626v1>

8. 5 Prominent AI Music Generation Models of Today - Data Science Dojo, accessed March 14, 2025, <https://datasciencedojo.com/blog/5-ai-music-generation-models/>

9. The Process of Developing an AI Music Generator: A Technical Overview - Medium, accessed March 14, 2025, <https://medium.com/@seekmeai/the-process-of-developing-an-ai-music-generator-a-technical-overview-5a736820f259>

10. How to Train Your ML Models: Music Generation (Try it out!) | by Yuehan | Medium, accessed March 14, 2025, <https://yuehan-z.medium.com/how-to-train-your-ml-models-music-generation-try-it-out-d4c0ab01c9f4>

11. Efficiently Fine-Tuning MusicGen for Text Conditioned Music Generation - Activeloop, accessed March 14, 2025, <https://www.activeloop.ai/resources/efficiently-fine-tuning-music-gen-for-text-conditioned-music-generation/>

12. audiocraft/docs/MUSICGEN.md at main - GitHub, accessed March 14, 2025, <https://github.com/facebookresearch/audiocraft/blob/main/docs/MUSICGEN.md>

13. MusicGen - Hugging Face, accessed March 14, 2025, <https://huggingface.co/docs/transformers/model_doc/musicgen>

14. TchAIkovsky – Piano MIDI Generation with Transformers, accessed March 14, 2025, <https://huggingface.co/blog/afmck/tchaikovsky>

15. Stable Audio Open, accessed March 14, 2025, <https://stable-audio-open.vercel.app/>

16. Stable Audio Open — Stability AI, accessed March 14, 2025, <https://stability.ai/news/introducing-stable-audio-open>

17. Jukebox Diffusion. An AI tool for conditional music… | by Jeff Sontag | Better Programming, accessed March 14, 2025, <https://medium.com/better-programming/jukebox-diffusion-cbe22ff3cd47>

18. Best model for music generation - Hugging Face Forums, accessed March 14, 2025, <https://discuss.huggingface.co/t/best-model-for-music-generation/133604>

19. How to Train Your Own Generative AI Model? Full Manual - Agente, accessed March 14, 2025, <https://agentestudio.com/blog/train-generative-ai-models>

20. How to Build an AI Model: A Step-by-Step Guide - Prismetric, accessed March 14, 2025, <https://www.prismetric.com/how-to-build-ai-model/>

21. The Future of AI in Music and Fine Art: A Creative Revolution - Art Plug, accessed March 14, 2025, <https://artplug.com/ai-in-music-and-fine-art/>

22. MusicGen - Advanced AI Music Generation, accessed March 14, 2025, <https://musicgen.com/>

23. Running Your Custom LoRA Fine-Tuned MusicGen Large Locally, accessed March 14, 2025, <https://huggingface.co/blog/theeseus-ai/musicgen-lora-large>

24. multimodal-art-projection/YuE: YuE: Open Full-song Music Generation Foundation Model, something similar to Suno.ai but open - GitHub, accessed March 14, 2025, <https://github.com/multimodal-art-projection/YuE>

25. Best AI Music Generator Software in 2025 - AudioCipher, accessed March 14, 2025, <https://www.audiocipher.com/post/ai-music-app>

26. How to use AI in music production - Style Transfer - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=3n9nqv2ZNig>

27. BassNet - AI that creates your personal bassline | Sony's TechHub, accessed March 14, 2025, <https://techhub.developer.sony.com/bassnet>

28. moslehi/deep-learning-music-style-transfer - GitHub, accessed March 14, 2025, <https://github.com/moslehi/deep-learning-music-style-transfer>

29. A Training-Free Approach for Music Style Transfer with Latent Diffusion Models - arXiv, accessed March 14, 2025, <https://arxiv.org/html/2411.15913v1>

30. Groove2Groove – One-shot music style transfer, accessed March 14, 2025, <https://groove2groove.telecom-paris.fr/>

31. MIDI Files as Training Data | Towards Data Science, accessed March 14, 2025, <https://towardsdatascience.com/midi-files-as-training-data-b67852c8b291/>

32. Applying Language Model Techniques to Compose AI Music | NVIDIA Technical Blog, accessed March 14, 2025, <https://developer.nvidia.com/blog/leveraging-ai-music-with-nvidia-dgx-2/>

33. Preparing Data for Machine Learning Models: Audio and Beyond. | by Yuehan - Medium, accessed March 14, 2025, <https://medium.com/@yuehan-z/preparing-data-for-machine-learning-models-audio-and-beyond-1b49daa16b0f>

34. A-Muze-Net: Music Generation by Composing the Harmony based on the Generated Melody, accessed March 14, 2025, <https://research.facebook.com/publications/a-muze-net-music-generation-by-composing-the-harmony-based-on-the-generated-melody/>

35. Multimodal Music Generation with Explicit Bridges and Retrieval Augmentation - arXiv, accessed March 14, 2025, <https://arxiv.org/html/2412.09428v1>

36. A How-To Guide: Creating Music with AI Music Generators - Soundful, accessed March 14, 2025, <https://soundful.com/how-to-guide-creating-music-with-ai-music-generators/>

37. AI & Creativity – #5: Controlled Inputs – The Key to High-Quality Output - Netgen, accessed March 14, 2025, <https://netgen.io/blog/ai-and-creativity_controlled-inputs_the-key-to-high-quality-output>

38. Music Generation | Qosmo - AI Creativity & Music Lab, accessed March 14, 2025, <https://qosmo.jp/en/products/music-generation>

39. Fine-tune MusicGen to generate music in any style - Replicate blog, accessed March 14, 2025, <https://replicate.com/blog/fine-tune-musicgen>

40. How To Train AI Voice Models ONLINE For FREE (No GPU Needed) - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=tnfqIQ11Qek>

41. Stability-AI/stable-audio-tools: Generative models for conditional audio generation - GitHub, accessed March 14, 2025, <https://github.com/Stability-AI/stable-audio-tools>

42. Top 10 AI Inference Platforms in 2025: Comparing LLM API Providers - Helicone, accessed March 14, 2025, <https://www.helicone.ai/blog/llm-api-providers>

43. Stable Diffusion 3.5 - Learn Prompting, accessed March 14, 2025, <https://learnprompting.org/docs/models/stable_diffusion_3_5>

44. Feature extraction — librosa 0.11.0 documentation, accessed March 14, 2025, <https://librosa.org/doc/main/feature.html>

45. Feature extraction — librosa 0.8.1 documentation, accessed March 14, 2025, <http://librosa.org/doc-playground/0.8.1/feature.html>

46. Write simple melodies with Music21 and Python - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=MYVWPmzsRz8>

47. music21.midi, accessed March 14, 2025, <https://www.music21.org/music21docs/moduleReference/moduleMidi.html>

48. Create an Audio Editor in Python using PyDub - GeeksforGeeks, accessed March 14, 2025, <https://www.geeksforgeeks.org/create-an-audio-editor-in-python-using-pydub/>

49. Manipulating audio files with PyDub | Python - DataCamp, accessed March 14, 2025, <https://campus.datacamp.com/courses/spoken-language-processing-in-python/manipulating-audio-files-with-pydub?ex=6>

50. How To Make AI-Generated Music? - Siteefy, accessed March 14, 2025, <https://siteefy.com/how-make-ai-generated-music/>

51. ylacombe/musicgen-dreamboothing: Fine-tune your own ... - GitHub, accessed March 14, 2025, <https://github.com/ylacombe/musicgen-dreamboothing>

52. DiffRhythm: Revolutionizing Open Source AI Music Generator - Hugging Face, accessed March 14, 2025, <https://huggingface.co/blog/Dzkaka/diffrhythm-open-source-ai-music-generator>

53. m-a-p/YuE-s1-7B-anneal-en-icl - Hugging Face, accessed March 14, 2025, <https://huggingface.co/m-a-p/YuE-s1-7B-anneal-en-icl>

54. Exploring the World of Open-Source Text-to-Speech Models - BentoML, accessed March 14, 2025, <https://www.bentoml.com/blog/exploring-the-world-of-open-source-text-to-speech-models>

55. 5 Best AI Singing Voice plugins and generators in 2023 - Controlla XYZ, accessed March 14, 2025, <https://www.controlla.xyz/post/5-best-ai-singing-voice-plugins-and-generators-in-2023>

56. 10 Best AI Singing Voice Generators for Making Music in 2025 - AudioCipher, accessed March 14, 2025, <https://www.audiocipher.com/post/ai-voice-generators>

57. Musicfy's Free AI Melody Generator, accessed March 14, 2025, <https://musicfy.lol/blog/ai-melody-generator>

58. Free Tool To Train AI Voice & Complete Step-by-Step Guide (Free, No Sign-Up, Unlimited) | Musicfy AI Blog, accessed March 14, 2025, <https://musicfy.lol/blog/train-ai-voice>

59. 7 Best Audio to MIDI Converter Software Tools For Musicians - AudioCipher, accessed March 14, 2025, <https://www.audiocipher.com/post/audio-to-midi>

60. An open source MIDI converter from Spotify - About - Basic Pitch, accessed March 14, 2025, <https://basicpitch.spotify.com/about>

61. 5 Ways to Use an AI Stem Splitter - LANDR Blog, accessed March 14, 2025, <https://blog.landr.com/ways-to-use-ai-stem-splitter/>

62. How Separate Stems | Music AI, accessed March 14, 2025, <https://music.ai/workflows/stem-separation-and-enhancement/stem-separation-suite/>

63. Fine-Tuning with MusicGen - MusicGen AI, accessed March 14, 2025, <https://musicgenai.org/musicgen-fine-tune/>

64. gabotechs/MusicGPT: Generate music based on natural language prompts using LLMs running locally - GitHub, accessed March 14, 2025, <https://github.com/gabotechs/MusicGPT>

65. Deep Learning for Music Composition | by Hey Amit | Data Scientist's Diary - Medium, accessed March 14, 2025, <https://medium.com/data-scientists-diary/deep-learning-for-music-composition-ed21ef1fccf7>

66. A.I. MIDIs is all you need! - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=wmMa8AsIlZ8>

67. Lemonaide: The World's First Fairly Trained AI MIDI Plugin, accessed March 14, 2025, <https://midi.org/lemonaide-the-worlds-first-fairly-trained-ai-midi-plugin>

68. 5 Open Source Generative Music Models You Can't Miss - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=GQfKoIMpea8>

69. Finetune a personal AI music generator - Lightning AI, accessed March 14, 2025, <https://lightning.ai/lightning-ai/studios/finetune-a-personal-ai-music-generator>

70. Ai Music Style Transfer Techniques | Restackio, accessed March 14, 2025, <https://www.restack.io/p/ai-music-generation-answer-style-transfer-cat-ai>

71. Ai Music Generation For Gen Z | Restackio, accessed March 14, 2025, <https://www.restack.io/p/ai-music-generation-answer-gen-z-cat-ai>

72. Audio Conditioning for Music Generation via Discrete Bottleneck Features - arXiv, accessed March 14, 2025, <https://arxiv.org/html/2407.12563v1>

73. Music Generation With MusicGen on an AMD GPU — ROCm Blogs, accessed March 14, 2025, <https://rocm.blogs.amd.com/artificial-intelligence/MusicGen/README.html>

74. A Guide to Generating Music using AudioCraft - Wandb, accessed March 14, 2025, <https://wandb.ai/geekyrakshit/audiocraft/reports/A-Guide-to-Generating-Music-using-AudioCraft--Vmlldzo1MzgwOTYz>

75. Harmonizing AI — Notes: Exploring MusicGen's Capabilities | by Joe El Khoury - Medium, accessed March 14, 2025, <https://medium.com/@jelkhoury880/musicgen-notes-8388202dc307>

76. MusicGen Melody - Hugging Face, accessed March 14, 2025, <https://huggingface.co/docs/transformers/model_doc/musicgen_melody>

77. transformers/docs/source/en/model\_doc/musicgen\_melody.md at main - GitHub, accessed March 14, 2025, <https://github.com/huggingface/transformers/blob/main/docs/source/en/model_doc/musicgen_melody.md>

78. facebook/musicgen-melody-large - Hugging Face, accessed March 14, 2025, <https://huggingface.co/facebook/musicgen-melody-large>

79. MusicGen: Simple and Controllable Music Generation, accessed March 14, 2025, <https://ai.honu.io/papers/musicgen/>

80. How to Use MusicGen for Music Generation | Edlitera, accessed March 14, 2025, <https://www.edlitera.com/blog/posts/musicgen>

81. MusicGen AI Architecture Explained: All You Need to Know, accessed March 14, 2025, <https://musicgenai.org/musicgen-architecture/>

82. facebook/musicgen-large - Hugging Face, accessed March 14, 2025, <https://huggingface.co/facebook/musicgen-large>

83. Stable Diffusion 3 Medium Fine-tuning Tutorial - Stability AI, accessed March 14, 2025, <https://stability.ai/learning-hub/stable-diffusion-3-medium-fine-tuning-tutorial>

84. Stable Audio Open 1.0 | Open Source\* Generative Audio and Fine Tuning\* - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=wDKIt3eQDHM>

85. Top 9 AI Music Generators to Try in 2025 - The Future of Music is Here - Kripesh Adwani, accessed March 14, 2025, <https://kripeshadwani.com/ai-music-generators/>

86. Home | MixAudio, accessed March 14, 2025, <https://mix.audio/>

87. Multimodal AI | Google Cloud, accessed March 14, 2025, <https://cloud.google.com/use-cases/multimodal-ai>

88. 10 AI Music Generators for Creators in 2025 | DigitalOcean, accessed March 14, 2025, <https://www.digitalocean.com/resources/articles/ai-music-generators>

89. Generate Songs in the Style of Your Favorite Artist Using AI - Blog - Soundraw, accessed March 14, 2025, <https://blog.soundraw.io/post/generate-songs-in-the-style-of-your-favorite-artist-using-ai>

90. The Power of AI in Music Creation: Why Loudly's AI Music Generator Is a Game-Changer for Creators - LALAL.AI, accessed March 14, 2025, <https://www.lalal.ai/blog/the-power-of-ai-in-music-creation-why-loudlys-ai-music-generator-is-a-game-changer-for-creators/>

91. MixAudio - Multimodal AI Music Generator - Lifetime Deal - DealFuel, accessed March 14, 2025, <https://www.dealfuel.com/seller/mixaudio-multimodal-ai-music-generator/>

92. MusicLM: Generating Music From Text, accessed March 14, 2025, <https://google-research.github.io/seanet/musiclm/examples/>

93. Ai Music Generator Huggingface | Restackio, accessed March 14, 2025, <https://www.restack.io/p/ai-music-generation-answer-huggingface-cat-ai>

94. Zeroxdesignart/music-ai - Hugging Face, accessed March 14, 2025, <https://huggingface.co/Zeroxdesignart/music-ai>

95. Best AI Music Extender: Turn Audio Samples Into Full Songs - AudioCipher, accessed March 14, 2025, <https://www.audiocipher.com/post/audio-to-audio>

96. AI Bass Generator: Unlocking the Secret Weapon - Empress, accessed March 14, 2025, <https://blog.empress.ac/ai-bass-generator-unlocking-the-secret-weapon-clldtg50d808813unb2nhl2e7w/>

97. Top Bass Completely Free AI Generators Recommendation - TopMediai, accessed March 14, 2025, <https://www.topmediai.com/ai-music/bass-completely-free-ai-generator/>

98. AI Bass Generator? Unison Bass Dragon - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=ZicdvJ-sxkk>

99. Fritz AI for Lens Studio: Training a Custom Style Transfer Model - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=efSDFwGk-ZU>

100. Github repository for inzva-ai project Audio Style Transfer, accessed March 14, 2025, <https://github.com/inzva/Audio-Style-Transfer>

101. A Training-Free Approach for Music Style Transfer with Latent Diffusion Models - arXiv, accessed March 14, 2025, <https://arxiv.org/abs/2411.15913>

102. 6 Best AI MIDI Generator DAW Plugins and Standalone Apps - AudioCipher, accessed March 14, 2025, <https://www.audiocipher.com/post/ai-midi-generators>

103. Create Original Music with AI MIDI Composer Tools - Staccato AI, accessed March 14, 2025, <https://staccato.ai/music>

104. Staccato's AI MIDI Maker | Create Original MIDI Music With the Help of AI, accessed March 14, 2025, <https://staccato.ai/midi-maker>

105. How to Generate Music with AI - Rootstrap, accessed March 14, 2025, <https://www.rootstrap.com/blog/how-to-generate-music-with-ai>

106. Dance2MIDI: Dance-Driven Milti-Instruments Music Generation, accessed March 14, 2025, <https://dance2midi.github.io/>

107. How To Edit MIDI Notes And Volumes Using Python With Music21 - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=1kPr1na1xtE>

108. music21.midi.translate, accessed March 14, 2025, <https://www.music21.org/music21docs/moduleReference/moduleMidiTranslate.html>

109. MIDI Music Data Extraction using Music21 and Word2Vec on Kaggle - Medium, accessed March 14, 2025, <https://medium.com/towards-data-science/midi-music-data-extraction-using-music21-and-word2vec-on-kaggle-cb383261cd4e>

110. Extracting audio features using Librosa | by Kaavya Mahajan - Medium, accessed March 14, 2025, <https://kaavyamaha12.medium.com/extracting-audio-features-using-librosa-3be4ff1fe57f>

111. Mel Spectrograms with Python and Librosa | Audio Feature Extraction - YouTube, accessed March 14, 2025, <https://www.youtube.com/watch?v=g8Q452PEXwY>

112. BirdCLEF:LIBROSA Audio Feature Extraction - Kaggle, accessed March 14, 2025, <https://www.kaggle.com/code/shreyasajal/birdclef-librosa-audio-feature-extraction>

113. Introduction to PyDub | Python - DataCamp, accessed March 14, 2025, <https://campus.datacamp.com/courses/spoken-language-processing-in-python/manipulating-audio-files-with-pydub?ex=1>