Music Generation from Brain Scans

Major Project submitted to

Faculty of Engineering and Technology, Jamia Millia Islamia

in partial fulfilment of the requirements for

The degree of

Bachelor of Technology

from

Department of Computer Engineering,

Faculty of Engineering and Technology

by

M. Abbas Ansari and Uzma Firoz Khan

under the supervision of

Prof. Tanvir Ahmad

Faculty of Engineering and Technology

Jamia Millia Islamia

Delhi-110025, India

June 2024

Declaration

We, M. Abbas Ansari and Uzma Firoz Khan, registered as undergraduate scholars, bearing Roll numbers 20BCS041, 20BCS052 for the Bachelor of Technology Programme under the Faculty of Engineering and Technology of Jamia Millia Islamia do hereby declare that We have completed the requirements as per mentioned by the university for project submission.

We do hereby declare that the project submitted is original and is the outcome of the independent investigations/research carried out by me and contains no plagiarism. The work is leading to the discovery of new techniques. This work has not been submitted by any other University or Body in quest of a degree, diploma or any other kind of academic award.

We do hereby further declare that the text, diagrams or any other material taken from other sources (including but not limited to books, journals and web) have been acknowledged, referred and cited to the best of our knowledge and understanding.

Ansari 21/5/2024

Mohammed Abbas Ansari (20BCS041)

21/05/2024

Uzma Firoz Khan (20BCS52)

Certificate

This is to certify that the project work entitled "Music Generation from Brain Scans" by Mohammed Abbas Ansari (20BCS041) and Uzma Firoz Khan (20BCS052) is a record of bonafide work carried out by them, in the Department of Computer Engineering, Jamia Millia Islamia, New Delhi, under my supervision and guidance in partial fulfilment of requirements for the award of Bachelor Of Engineering in Computer Engineering, Jamia Millia Islamia in the academic year 2024.

21.5.24

Prof. Dr. Bashir Alam
(Head of Department)
Department of Computer Engineering,
Faculty of Engineering and Technology
Jamia Millia Islamia
New Delhi

21/5/24

Prof. Dr. Tanvir Ahmad
(Professor)
Department of Computer Engineering,
Faculty of Engineering and Technology
Jamia Millia Islamia
New Delhi

Abstract

This project explores the fascinating possibility of generating music from brain activity recorded during functional magnetic resonance imaging (fMRI) scans, bridging the gap between neuroscience, artificial intelligence, and music composition. Our innovative approach employs the Map method to learn a mapping between the fMRI response tensor and the prior embedding space of a conditional music generation model, MusicGen. By experimenting with different modality encoders, including EnCodec, Chromagram Tokenizer, and T5, and temporal alignment techniques, such as sliding window averaging, skipped timesteps, and total averaging, we demonstrate the feasibility of reconstructing music from brain scans. The text-based T5 encoder emerges as the most effective modality, while the total averaging technique proves to be the most successful in aligning the fMRI response tensor with the prior embedding space. We identify the top-performing regions of interest (ROIs) in the brain for music generation, primarily located in the temporal lobe and associated with auditory processing, language comprehension, and multimodal integration. The evaluation of the generated music samples using objective metrics, such as Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD), showcases the challenges and limitations of the current approach, but also highlights the potential for future advancements. This research contributes to a deeper understanding of the human brain and its relationship with the arts, inspiring new forms of musical creation and fostering collaborations between neuroscientists, musicians, and AI researchers. As we continue to explore the intersection of the mind, music, and machine, we move closer to a future where the boundaries between these domains dissolve, giving rise to new forms of creative expression and artistic exploration.

Keywords: Music generation, brain scans, fMRI, artificial intelligence, neuroscience, MusicGen, Map method, auditory processing, language comprehension, multimodal integration,

Acknowledgements

We extend our sincere appreciation to Dr. Tanvir Ahmad, Professor in the Department of Computer Engineering at Jamia Millia Islamia, New Delhi, for his invaluable technical guidance, insightful innovative ideas, and unwavering support throughout our minor research thesis. His expertise has been instrumental in shaping the direction of our project, and We are truly grateful for his mentorship. We also express our gratitude to Prof. Bashir Alam, Head of the Department, whose consistent support has been a pillar of strength during the entire research process. We are appreciative of the Department of Computer Engineering and the entire faculty for their commitment to teaching, guidance, and encouragement, which have been pivotal to our academic journey. Furthermore, We extend our thanks to our classmates and friends for their valuable suggestions and support whenever required. We acknowledge the collaborative efforts and contributions of everyone involved, regretting any inadvertent omissions.

M. Abbas Ansari and Uzma Firoz Khan

Contents

List of Figures

List of Tables

Abbreviations

JMI	Jamia Millia Islamia
CNN	Convolutional Neural Networks
MRI	Magnetic Resonance Imaging
MEG	Magnetoencephalography
fMRI	functional Magnetic Resonance Imaging
BOLD	Blood Oxygenation Level Dependent
ROI	region of interest
CSF	cerebrospinal fluid
BBR	Boundary-Based Registration
RVQ	Residual Vector Quantization
FAD	Fréchet Audio Distance
KL	Kullback-Leibler divergence
AIGC	AI Generated Content
MFCC	mel-frequency cepstral coefficients
MCD	Mel Cepstral Distortion

Chapter 1

Introduction

Music is a universal language that transcends cultural boundaries, serving as a powerful medium for expression and communication. The neural representations of music in the human brain have been a topic of great interest in the field of neuroscience. Numerous studies have investigated brain activity using functional magnetic resonance imaging (fMRI) while participants listened to music, revealing the representation of various musical features such as rhythms [13], timbres [14, 15], emotions [16], and musical genres [17, 18]. These findings provide valuable insights into the complex nature of human music perception and experience.

Recent advancements in text-to-music models have enabled the conditional generation of high-quality music [19, 20, 21, 22, 23]. This development has opened up new possibilities for bridging the gap between our linguistic understanding of music and the creation of musical compositions. However, the relationship between the text and music embeddings used in these generative models and the neural representations of music in the human brain remains largely unexplored. Furthermore, the potential for generating music directly from brain activity has yet to be fully investigated.

Decoding brain signals to understand perception is crucial for unraveling the intricate mechanisms of human perception [24, 25]. The correlation between external stimuli and neural responses provides valuable insights into the underlying perceptual processes [26, 27, 28]. Perception involves not only the passive reception of sensory inputs but also complex cognitive processes that shape subjective experiences. Therefore, decoding brain signals back into corresponding perceptual modalities, such as vision, audio, and descriptive semantic text, holds significant implications for understanding the neural basis of perception. Moreover, brain perception decoding contributes to the development of practical Brain-Computer Interface (BCI) systems, which have the potential to enhance communication between the brain and external devices, with applications in neuroprosthetics [29, 30], virtual/augmented reality [31, 32], and assistive technologies [33, 34].

Neuroimaging techniques, such as fMRI, Electroencephalography (EEG), and Magnetoencephalography (MEG), provide a window into the neural activity associated with perceptual experiences [35, 36, 37]. Each technique offers unique advantages in terms of spatial and temporal resolution, allowing for a comprehensive understanding of brain function. Neuroimaging data serves as the foundation for brain decoding, enabling the exploration of the functionalities and interrelations of various brain regions and shedding light on the mechanisms underlying perception and comprehension [38, 39].

Recent studies have revealed correspondences between the internal representations of deep learning models and those of the brain across various sensory and cognitive modalities [40, 41]. These findings have advanced our understanding of brain functions through the development of encoding models [42], interpretations of representations based on their correspondence with brain functions [43, 44], and the reconstruction of experienced content from brain activity [45, 46, 47]. In the context of auditory brain functions, researchers have developed encoding models using deep learning models that process auditory inputs [41] and have conducted studies to reconstruct perceived sounds from brain activity [48, 49].

Building upon the foundational work in the field, we have developed a novel approach to reconstruct music from fMRI recordings of subjects listening to music. Our methodology utilizes the music genre neuroimaging dataset from Nakai et al. (2022) [50], which provides fMRI scans along with corresponding music stimuli. We employ advanced techniques, such as linear regression and ensemble modeling, to predict music embeddings from voxel data, considering various temporal alignment strategies to address the challenges posed by the different temporal resolutions of fMRI scans and music embeddings. A key aspect of our work is the exploration of different modality encoders, including EnCodec for audio waveforms, Chromagram Tokenizer for melodic representations, and T5 for textual descriptions of music. By comparing the performance of these encoders in predicting music embeddings from brain activity, we provide valuable insights into the most effective modalities for capturing the neural representations of music.

Our research also delves into the identification of the top-performing regions of interest (ROIs) in the brain for music generation. Through rigorous analysis, we have discovered that the temporal lobe and its associated regions, such as the superior temporal gyrus and the inferior temporal gyrus, play a crucial role in music perception and generation. These findings contribute to a deeper understanding of the neural mechanisms underlying music processing and shed light on the complex interplay between different brain regions in the experience of music.

To evaluate the quality of the generated music, we employ objective metrics such as Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD). Our results demonstrate the effectiveness of our approach, with the text-based T5 encoder and the total averaging technique achieving the best performance among the evaluated methods. While there is still room for improvement in terms of the fidelity of the generated music compared to the ground truth, our work represents a significant step forward in the field of brain-based music generation.

Chapter 2

Related Work

2.1 Music Generation

The representation of audio signals is a crucial component in generative models for music. The prominent approach is to represent music signals in a compressed representation, either discrete or continuous, and apply a generative model on top of it. Lakhotia et al. [51] proposed quantizing speech representations using k-means to construct speech language models. Recently, EnCodec[5] and Soundstream [52] proposed applying a VQ-VAE directly on the raw waveform using residual vector quantization. Such discrete representations have been used for various audio generation tasks, including text-to-audio generation. Music generation has been a long-standing area of research, with various approaches explored. MuseGAN[53] proposed a GAN-based approach for symbolic music generation, while Bassan et al.[54] introduced an unsupervised segmentation method for symbolic music, which can be used for generation. Ycart et al.[55] modeled polyphonic music using recurrent neural networks, and Ji et al.[56] conducted a comprehensive survey on deep learning methods for music generation. Autoregressive models have been widely used for music generation. Jukebox[57] proposed representing music samples in multiple streams of discrete representations using a hierarchical VQ-VAE, and applied sparse transformers over the sequences to generate music. While Jukebox generates music with high temporal coherence, it contains perceptible artifacts. Gan et al.[58] focused on generating music for a given video, predicting MIDI notes. More recently, Agostinelli et al.[59] represented music using "semantic tokens" and "acoustic tokens", and employed a cascade of transformer decoders conditioned on a textual-music joint representation, Mulan[60]. Donahue et al.[61] followed a similar modeling approach for singing-to-accompaniment generation. Diffusion models have emerged as an alternative approach for music generation. [62, 63, 64, 65] proposed using latent diffusion models for the task of text-to-music generation. Schneider et al.[62] used diffusion models for both audio encoder-decoder and latent generation, while Huang et al.[63] proposed a cascade of diffusion models to generate audio and gradually increase its sampling rate. Forsgren and Martiros[65] fine-tuned Stable Diffusion Rombach et al.[66] using spectrograms to generate five-second segments, then employed image-to-image mapping and latent interpolation to generate longer sequences. In the broader domain of audio generation, several studies have focused on **text-to-audio** generation for environmental sounds. Yang et al.[67] represented audio spectrograms using a VQ-VAE, then applied a discrete diffusion model conditioned on textual CLIP embeddings Radford et al.[68] for generation. Kreuk et al.[69] proposed applying a transformer language model over discrete audio representations obtained by quantizing time-domain signals using EnCodec[70]. Sheffer and Adi[71] followed a similar approach to Audiogen[69] for image-to-audio generation. Make-an-audio[72] and Audioldm[73] proposed using latent diffusion models for text-to-audio generation, extending it to various tasks such as inpainting and image-to-audio conversion.

2.2 Neural Decoding

Recent advancements in neuroimaging technologies, such as functional Magnetic Resonance Imaging (fMRI), Electroencephalography (EEG), and Magnetoencephalography (MEG), have opened up exciting possibilities in the field of neural decoding [35, 36, 37]. By analyzing patterns of brain activity, researchers aim to reconstruct the original stimuli that elicited those patterns, shedding light on how the brain processes and represents information [38, 39]. This section explores the current landscape of neural decoding, with a particular focus on reconstructing visual and auditory stimuli from brain scans.

2.2.1 Visual Decoding

The human visual cortex is a complex network of regions responsible for processing visual information [74, 75]. Several studies have investigated the decoding of visual stimuli from fMRI data, leveraging datasets such as the Natural Scenes Dataset (NSD) [76], Generic Object Decoding (GOD) dataset [77], and Deep Image Reconstruction (DIR) dataset [78]. These datasets contain fMRI scans of subjects viewing a variety of natural images, enabling the exploration of both early and higher visual cortex regions.

Researchers have employed various generative models to reconstruct images from brain activity. For instance, BrainSD [79] and BrainDiffuser [80] utilize diffusion models conditioned on fMRI data to generate realistic images. GANs have also been popular, with models like BrainSSG [81] and BrainDVG [82] leveraging adversarial training to improve the quality of reconstructed images. Additionally, autoencoder-based approaches, such as SSNIR [83] and SSNIR-SC [84], have demonstrated promising results in capturing low-level visual details.

EEG-based visual decoding has also gained attention, with datasets like EEG-VOA [85] enabling the study of object recognition from EEG signals. Models such as DreamDiffusion [86] and NeuroImagen [87] have pushed the boundaries of EEG-based image reconstruction, generating remarkably detailed and semantically coherent images.

2.2.2 Auditory Decoding

Decoding auditory stimuli from brain activity is an exciting frontier in neural decoding. The Brain Sound Reconstruction (BSR) dataset [88] and Narratives dataset [89] provide valuable resources for studying the reconstruction of sound and speech from fMRI data. BSR [88] utilizes an autoregressive approach to generate audio signals based on brain activity patterns, while works like UniCoRN [90] focus on reconstructing textual descriptions of spoken stories.

EEG-based auditory decoding has also shown promise, with datasets like ETCAS [91] enabling the study of continuous speech reconstruction from EEG signals. The ETCAS model [91] employs a GAN-based approach to directly map EEG signals to speech waveforms, showcasing the potential for real-time speech decoding.

2.2.3 Multimodal Decoding

The human brain processes information from multiple sensory modalities, and recent work has explored the simultaneous decoding of different stimuli types. The Continous Language Semantic Reconstruction (CLSR) dataset [92] contains fMRI data of subjects viewing silent video clips and listening to spoken stories, allowing for the investigation of multimodal decoding. The CLSR model [92] demonstrates the ability to generate textual descriptions from both video and speech-evoked brain activity. Another exciting avenue is the decoding of music from brain scans. Datasets like MusicGenre [93] and MusicAffect [94] provide fMRI and EEG data of subjects listening to various musical stimuli. Models such as

Brain2Music [95] and NDMusic [96] have shown promising results in reconstructing music from brain activity, opening up possibilities for brain-computer interfaces in the musical domain.

Chapter 3

Theoretical Background

3.1 Audio Data Representation

The advent of deep learning algorithms has prompted many researchers to explore alternatives to traditional signal processing methods for sound generation. While deep learning models have demonstrated remarkable capabilities in expressive voice synthesis, realistic sound texture generation, and virtual instrument note synthesis, the quest for the most suitable deep learning architecture remains an active area of investigation. The choice of architecture is intricately tied to the representation of audio data. Raw audio waveforms, with their intrinsic density and richness, can pose challenges for deep learning models in terms of computational efficiency and training time. Moreover, the waveform representation may not align with the perceptual aspects of sound. Consequently, researchers have explored transforming raw audio into compressed and more

3.1.1 Raw Audio Waveform Representation and Quantization

The term ”raw audio” commonly refers to the waveform representation encoded using pulse code modulation (PCM), which involves sampling the continuous waveform in both time and amplitude domains. This process results in a sequence of numbers, each representing an amplitude value at a chosen sampling frequency. To ensure faithful reproduction, the highest frequency component must adhere to the Nyquist-Shannon sampling theorem [97], stating that frequencies below half the sampling rate can be accurately reconstructed. Typical audio applications employ a 44.1 kHz sampling frequency, with quantization levels ranging from 8 bits (256 levels) to 24 bits (16.8 million levels). Consequently, a one-second

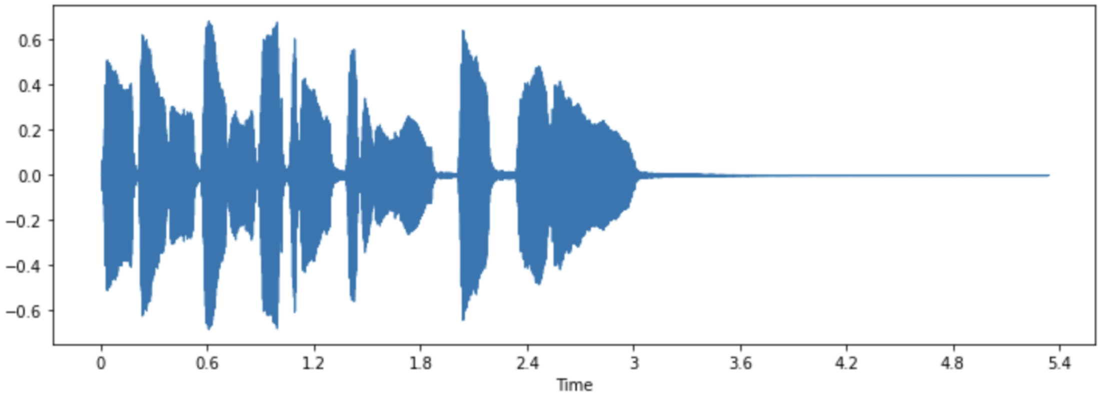


FIGURE 3.1: Sounds visualized as a waveform, which plots the sample values over time and illustrates the changes in the sound's amplitude. This is also known as the time domain representation of sound.

audio segment sampled at 44.1 kHz generates 44,100 samples, rendering this representation highly informative for deep learning models.

However, preprocessing techniques can enhance the effectiveness of deep learning models by reducing the quantization range. Several research approaches [98] [99] [100] employ non-linear quantization processes like -law companding, described by the equation:

$$f(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 < x < 1 \tag{3.1}$$

where μ represents the number of levels created after transformation. Despite the attention garnered by non-linear quantization methods, the predominant approach in existing literature [101] involves using a normalized high-resolution signal as input to deep learning models.

Alternatively, some applications adopt linear quantization of the input waveform [102] [103], maintaining a uniform quantization step size across the amplitude range. Furthermore, researchers have explored quantization designs that treat the most and least significant bits differently [104], potentially enhancing the representation of relevant signal characteristics.

3.1.2 Quantization and Preprocessing Considerations

The choice of quantization technique and preprocessing steps applied to the raw audio waveform can significantly impact the performance and effectiveness of deep learning models for audio generation tasks. Factors influencing this choice include the specific audio task, the deep learning architecture employed, and the trade-offs between computational complexity, reconstruction quality, and perceptual aspects of the generated audio.

3.1.3 Spectrogram Representations

Spectrograms provide a time-frequency visual representation of sound, enabling the analysis of spectral content evolution over time. They are typically obtained through the Short-Time Fourier Transform (STFT), which applies the Discrete Fourier Transform (DFT) to overlapping segments of the waveform, as described by the equation:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\omega kn}, \quad k = 0, 1, \ldots, N-1 \tag{3.2}$$

where N is the number of samples, and k is the segment index. The spectrogram employs only the absolute values of the STFT, discarding the phase information. This representation has been widely adopted in various audio-related research works[105] [106] [107].

In addition to the conventional spectrogram, deep learning architectures have explored non-linear variants, such as **mel-spectrograms** [108] [109][110][111][112][113][114] and Constant-Q Transformations (CQT) [115]. The mel-spectrogram is generated by applying perceptual mel-filter bands to the DFT, with the most common encoding formula given by:

$$\text{mel} = 2595 \log_{10}(1 + f/700) \tag{3.3}$$

where f is the frequency in Hertz. Alternatively, some models capture the perceptual transformation by applying a linear scaling up to 1 kHz and a logarithmic scaling above this threshold.

The CQT is another time-frequency representation with geometrically spaced frequencies, where the center frequencies of the filters are calculated as $\omega_k = 2^{k/b}\omega_0$, with

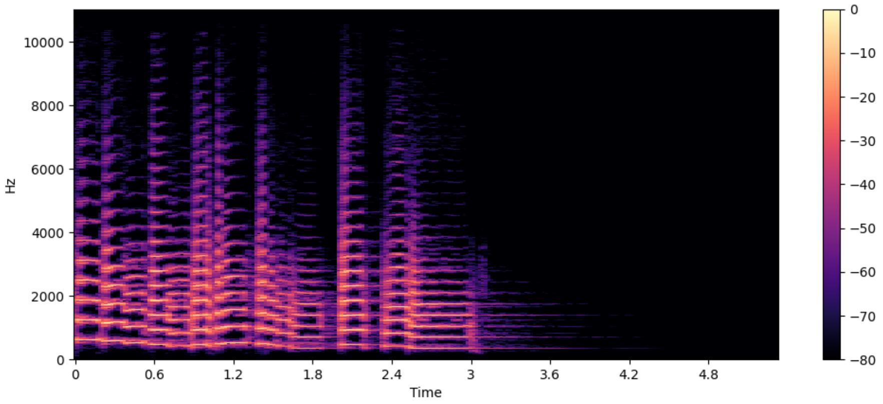


FIGURE 3.2: Mel spectrogram is a variation of the spectrogram that is commonly used in speech processing and machine learning tasks.

$k = 1, 2, \ldots, k_{\max}$ and b as a constant. The bandwidth of each frequency is given by $\delta_k = \omega_k(2^{1/b} - 1)^{-1}$, and the frequency resolution is determined by the quality factor Q

$$\mathrm{Q} = \omega_k/\delta_k = (2^{1/b} - 1)^{-1} \tag{3.4}$$

While the CQT offers different frequency resolutions for low and high frequencies, it discards the phase information and is often irreversible.

To address this limitation, researchers have explored invertible CQT variants based on nonstationary Gabor frames [116] and rainbowgrams [117], which encode time derivatives of the phase using colors.

Furthermore, more complex spectrogram-based representations have been investigated, such as scaled logarithmic amplitude and phase of the STFT, increased resolution spectrograms, mel-filtered spectrograms, and Instantaneous Frequency-based spectrograms [118]. Comparative studies between raw audio and spectrogram representations have also been conducted to uncover the most suitable representation for specific deep learning models [119] [120].

3.1.4 Acoustic Feature Representations

To overcome the wealth of acoustic information present in raw audio waveforms, various studies have explored the extraction of perceptual features from the original signal. These acoustic feature representations aim to capture salient characteristics in a more compact and interpretable form. Some approaches employ phoneme inputs [121], fundamental frequency and spectral features [122], or a combination of multiple attributes such as velocity, instrument, pitch, and time information [123]. Other implementations leverage cepstral coefficients [124] [125] or a variety of linguistic and acoustic features [126] [100]. Additionally, widely recommended parameter sets derived from the WORLD vocoder [127][128][129], a prominent analysis and synthesis tool for speech and audio signals, have been utilized.

The adoption of acoustic feature representations offers several advantages over raw waveform data, including reduced dimensionality, enhanced interpretability, and potential alignment with perceptual qualities of sound. However, the specific choice of feature set depends on factors such as the target audio application, the availability of prior domain knowledge, and the trade-offs between computational complexity, reconstruction quality, and the desired level of abstraction.

3.1.5 Embedding Representations

Inspired by their success in Natural Language Processing (NLP), embedding representations have been adopted in sound processing to encode audio signals into real-valued vectors. This approach leverages the property of similar embeddings being clustered in the vector space, enabling the encoding of analogous audio characteristics. Embeddings have been employed for various purposes, including:

- Reducing the dimensionality of audio signals [130] [109]
- Enhancing timbre synthesis [131]
- Generating interpretable representations [132][133] for effective parameter extraction in synthesizers

Certain architectures, such as autoencoders [117] and multi-resolution encoders [57], utilize embeddings as latent representations to condition deep learning models for audio generation tasks.

3.1.6 Symbolic Representations

In music processing, symbolic representations refer to the use of formats such as Musical Instrument Digital Interface (MIDI) and piano rolls. MIDI is a technical standard that specifies a protocol, a digital interface, and a communication link for the simultaneous operation of multiple electronic musical instruments. A MIDI file encodes the notes being played at each time step, including information about the instrument, pitch, and velocity. Prominent implementations like MidiNet [134] leverage MIDI data for music generation tasks.

Piano rolls offer a more dense representation of musical information compared to MIDI. A piece of music is represented as a binary $N \times T$ matrix, where N is the number of playable notes, and T is the number of time steps. Generative Adversarial Networks (GANs) have been applied to music generation using multiple-track piano-roll representations [53]. Certain approaches, such as DeepJ [135], scale the representation matrix between 0 and 1 to capture note dynamics.

However, a notable limitation of symbolic representations is their inability to differentiate between holding a note and replaying a note, as both are represented identically. To address this issue, DeepJ introduced a secondary ”replay” matrix alongside the original ”play” matrix.

The choice between embedding, symbolic, or other representations depends on factors such as the desired level of abstraction, computational complexity, interpretability, and fidelity to the original audio signal.

3.1.7 Chromagram Representation

The chromagram is a representation aimed at increasing the robustness of the log-frequency spectrogram to variations in timbre and instrumentation. The main idea is to combine pitch bands corresponding to pitches that differ by one or several octaves, leveraging the human perception of pitch periodicity. A pitch can be separated into two components: tone height, referring to the octave number, and chroma, representing the pitch spelling attribute contained in the set $\{C, C^{\sharp}, D, D^{\sharp}, \ldots, B\}$.

Enumerating the chroma values from 0 to 11, where 0 refers to C, 1 to $C^{\sharp}$, and so on, a pitch class is defined as the set of all pitches that share the same chroma. The chroma features aggregate all spectral information related to a given pitch class into a single coefficient. Given a pitch-based log-frequency spectrogram $\gamma_{LF} : \mathbb{Z} \times [0 : 127] \rightarrow \mathbb{R}_{\geq 0}$, a chromagram $\mathbb{Z} \times [0 : 11] \rightarrow \mathbb{R}_{\geq 0}$ can be derived by summing up all pitch coefficients that belong to the same chroma:

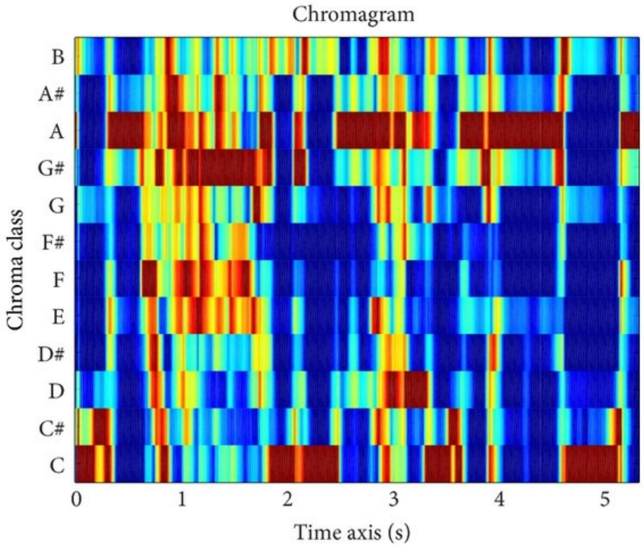


Figure 3.3: Example of chromagram from a piece of music.[1]

$$C(n,c) := \sum_{\substack{p \in [0:127] \\ p \bmod 12 = c}} \gamma_{LF}(n,p), \quad c \in [0:11] \tag{3.5}$$

The cyclic nature of chroma features becomes evident when visualizing the chromagram of a chromatic scale, where the increasing notes are "wrapped around" the chroma axis. However, due to the presence of higher harmonics, the energy is typically spread across various chroma bands even when playing a single note, as harmonics contribute to other chroma bands.

The chromagram representation leverages the human perception of pitch periodicity and octave equivalence, providing a more robust representation against variations in timbre and instrumentation compared to the log-frequency spectrogram.

3.2 Machine Learning

3.2.1 Linear Regression

Linear regression is a fundamental statistical and machine learning model employed for predicting a continuous target variable based on one or more input features. It assumes a linear relationship between the input variables and the output variable, making it a powerful yet interpretable model for various regression tasks.

3.2.1.1 Linear Regression and Regularization

Linear regression is a widely used algorithm for predicting numeric values based on a linear combination of input variables. The hypothesis for linear regression can be represented as:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

where $h_\theta(x)$ is the predicted output, x_i are the input variables, and θ_i are the model coefficients. The coefficients θ_i are typically learned by minimizing the **residual sum of squares (RSS)**:

$$\mathrm{RSS}(\theta) = \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))^2$$

where m is the number of training examples, $y^{(i)}$ is the true output, and $x^{(i)}$ is the input vector for the i^{th} training example. **Overfitting** occurs when the model becomes too complex and captures the noise in the training data, resulting in poor generalization to new, unseen data. One way to mitigate overfitting is through **regularization**, which involves adding a penalty term to the optimization objective to discourage complex models. There are two common types of regularization techniques: **Lasso Regularization (L1)** and **Ridge Regularization (L2)**

3.2.1.2 Linear Regression with L2 Regularization (Ridge Regression)

Ridge regression, or L2 regularization, is particularly useful when dealing with **multicollinearity**, which occurs when two or more predictors have a near-linear relationship. In such cases, the ordinary least squares (OLS) estimator may return erroneously high-value coefficients, leading to overfitting and unstable models. To address this issue, ridge regression introduces an L2 penalty term, also known as the **ridge penalty**, to the RSS function:

$$\text{Ridge Penalty} = \alpha \sum_{j=1}^{n} \theta_j^2$$

The **ridge regression estimator** minimizes the RSS function with the added ridge penalty:

$$\underset{\theta}{\text{minimize}}; \mathrm{RSS}(\theta) + \alpha \sum_{j=1}^{n} \theta_j^2$$

By adding the ridge penalty, the model coefficients are shrunk towards zero, but not all coefficients are shrunk by the same value. Instead, coefficients are shrunk in proportion to their initial size. As the regularization parameter α increases, high-value coefficients shrink at a greater rate than low-value coefficients. This process is known as **coefficient shrinkage**.

3.2.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that have proven to be highly effective in tasks such as image classification and object detection. CNNs are designed to extract increasingly abstract features from input data through a series of layers. The basic building blocks of a CNN are convolutional layers and pooling layers, followed by fully connected layers for classification or regression.

The feature learning process in CNNs involves two main components: convolutional layers and pooling layers.

3.2.2.1 Convolutional Layer

The convolutional layer is used for feature extraction. It applies a convolution operation followed by an activation function to the input data. Let I be the input tensor of dimensions $m_1 \times m_2 \times m_c$, where m_1 and m_2 represent the spatial dimensions, and m_c represents the number of channels. A kernel or filter K of dimensions $n_1 \times n_2 \times n_c$ (where n_c is the same as m_c) is convolved with the input tensor I. The filter moves over the input tensor from left to right, and the dot product between the filter and the corresponding input region is computed and summed up. The stride s determines the step size by which the filter moves over the input tensor. The resulting feature map has dimensions $(m_1 - n_1 + 1) \times (m_2 - n_2 + 1) \times 1$.

$$\mathrm{F}[i, j] = (I * K)_{[i,j]} \tag{3.6}$$

The (i, j)-th entry of the feature map is given by:

$$\mathrm{f}_{[i,j]} = \sum_{x}^{m_1} \sum_{y}^{m_2} \sum_{z}^{m_c} K_{x,y,z} \cdot I_{i+x-1,j+y-1,z} \tag{3.7}$$

Zero **padding** is often used to ensure that the filter can be applied to the corners of the input tensor, preventing information loss. In general , one bias term 'b' has been added to the convoluted part and then the activation function is applied.

$$\text{Conv}(I, K) = \phi_a(c) = \phi_a(I * K + b) \tag{3.8}$$

$$\text{where } \phi_a \text{ is an activation function.} \tag{3.9}$$

There are different types of activation functions as sigmoid, tangent, hyperbolic tangent function. The most commonly used activation function is ReLU which eliminates the negative values: such as the Rectified Linear Unit (ReLU):

$$\text{ReLU}(x) = \max(0, x) \tag{3.10}$$

3.2.2.2 Pooling Layer

The pooling layer is used to reduce the spatial dimensions of the feature maps, effectively down-sampling the output of the convolutional layers. Common pooling operations include max pooling, average pooling, and sum pooling.

$$\begin{aligned} \text{Conv}(I, K) &= c \\ P &= \phi_p(c) \end{aligned} \tag{3.11}$$

$$\text{where } \phi_p \text{ is a pooling function.} \tag{3.12}$$

3.2.2.3 Classification

After passing through multiple convolutional and pooling layers, the output is flattened into a single vector, which serves as the input to the fully connected layers.

3.2.2.4 Fully Connected Layer

The fully connected layer receives the flattened vector and performs classification or regression on the extracted features. The output of the fully connected layer is calculated as:

$$X = \sum_i w_i P_i + b' \tag{3.13}$$

$$z = g(X) \tag{3.14}$$

where g is the activation function of the fully connected layer.

Multiple fully connected layers can be stacked, with the output of one layer serving as the input to the next layer.

3.2.3 Transformers

Transformers, introduced by Vaswani et al [2] are a type of neural network architecture that has been widely adopted in various natural language processing (NLP) tasks, and more recently, in music generation tasks. The key innovation of Transformers lies in their reliance on self-attention mechanisms, which allow the model to capture long-range dependencies within the input data more effectively than traditional recurrent neural networks (RNNs) or convolutional neural networks (CNNs).

3.2.3.1 Position Embeddings:

Position embeddings play a crucial role in Transformer architectures, addressing a limitation inherent in self-attention mechanisms: the lack of inherent sequential information. Unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), Transformers do not inherently understand the sequential order of the input tokens, as they process the entire sequence in parallel. To capture the sequential order, position embeddings are introduced. The position embeddings are added to the input token embeddings to create enriched representations that encode both the token identity and its position.

3.2.3.2 Multi-Head Self Attention Mechanism:

The self-attention mechanism computes a weighted sum of input representations, where the weights are determined by the compatibility scores (attention weights) between different positions in the input sequence. The multi-head attention allows the model to attend to different positions in parallel.

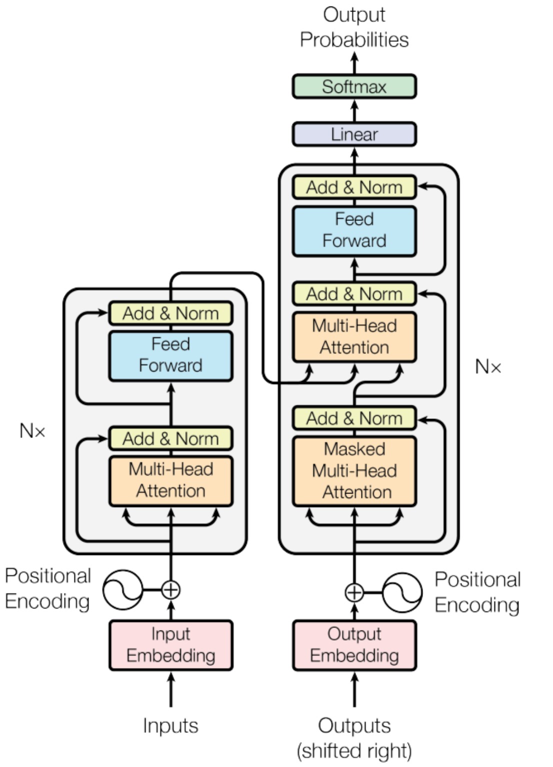


FIGURE 3.4: The Transformer - model architecture. [2]

The self-attention function is defined as follows for a single head:

$$Attention(Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{3.15}$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimension of key vectors. For multiple heads, the outputs are concatenated and linearly transformed.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Attention}_1(Q, K, V), \ldots, \text{Attention}_h(Q, K, V))W_O \tag{3.16}$$

3.2.3.3 Encoder

The encoder in the Transformer architecture is composed of a stack of $N = 6$ identical layers. Each layer consists of two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Residual connections [136] are employed around each sub-layer, followed by layer normalization [137]. The output of each sub-layer is given by:

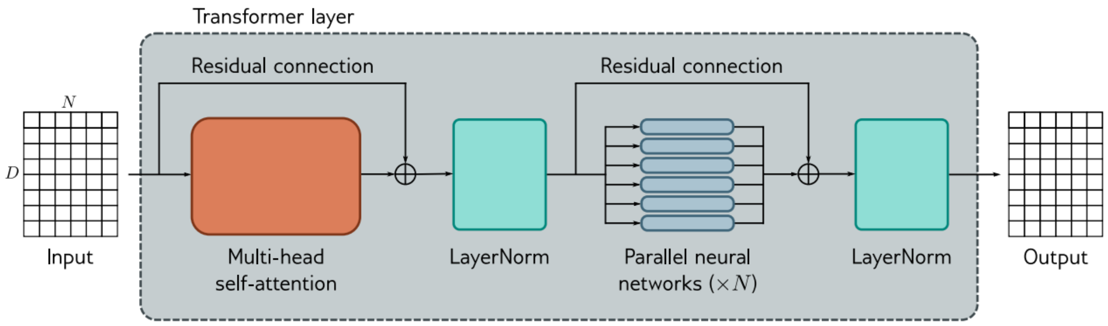


FIGURE 3.5: The transformer consists of a series of operations. Multi-head attention block and a LayerNorm operation is applied. A second residual layer where the same fully connected neural network is applied separately to each of the N representations. Finally, LayerNorm is applied again. [3]

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{3.17}$$

where x is the input to the sub-layer and Sublayer(x) is the function implemented by the sub-layer itself. All sub-layers, as well as the embedding layers, produce outputs of dimension $d_{\text{model}} = 512$.

3.2.3.4 Decoder

The decoder is also composed of a stack of $N = 6$ identical layers. In addition to the two sub-layers present in the encoder (multi-head self-attention and position-wise feed-forward network), the decoder introduces a third sub-layer that performs multi-head attention over the output of the encoder stack. Similar to the encoder, residual connections and layer normalization are applied after each sub-layer.

To maintain the auto-regressive property, the self-attention sub-layer in the decoder is modified to prevent positions from attending to subsequent positions. This masking, combined with the fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i.

3.2.4 T5 Model

The T5 model [4] is an encoder-decoder Transformer implementation that closely follows the originally proposed form by [2]. The input sequence of tokens is first mapped to a sequence of embeddings, which is then passed into the encoder. The encoder consists of a stack of blocks, each comprising a self-attention layer followed by a small feed-forward network. Layer normalization [137] is applied to the input of each subcomponent, using a simplified version where the activations are only rescaled without an additive bias. After layer normalization, a residual skip connection [136] adds each subcomponent's input to its output. Dropout [138] is applied within the feed-forward network, on the skip connection, on the attention weights, and at the input and output of the entire stack.

The decoder is similar in structure to the encoder, but includes a standard attention mechanism after each self-attention layer that attends to the output of the encoder. The self-attention mechanism in the decoder also uses a form of autoregressive or causal self-attention, which only allows the model to attend to past outputs. The output of the final decoder block is fed into a dense layer with a softmax output, whose weights are shared with the input embedding matrix. All attention mechanisms in the Transformer are split up into independent "heads" whose outputs are concatenated before being further processed.

Since self-attention is order-independent (i.e., an operation on sets), it is common to provide an explicit position signal to the Transformer. While the original Transformer used a sinusoidal position signal or learned position embeddings, the T5 model uses relative position embeddings [139][140]. Instead of using a fixed embedding for each position, relative position embeddings produce a different learned embedding according to the offset between the "key" and "query" being compared in the self-attention mechanism. The T5 model uses a simplified form of position embeddings where each "embedding" is simply a scalar added to the corresponding logit used for computing the attention weights. For efficiency, the position embedding parameters are shared across all layers, though within a given layer, each attention head uses a different learned position embedding. A fixed number of embeddings (32 in the T5 model) are learned, each corresponding to a range of possible key-query offsets, with ranges increasing logarithmically up to an offset of 128, beyond which all relative positions are assigned to the same embedding.

The T5 model is roughly equivalent to the original Transformer proposed by Vaswani et al. [2], with the exception of removing the Layer Norm bias, placing the layer normalization outside the residual path, and using the different position embedding scheme described above.

To study the scalability of these models, [4] experiment with how performance changes as the models are made to have more parameters or layers. Training large models can

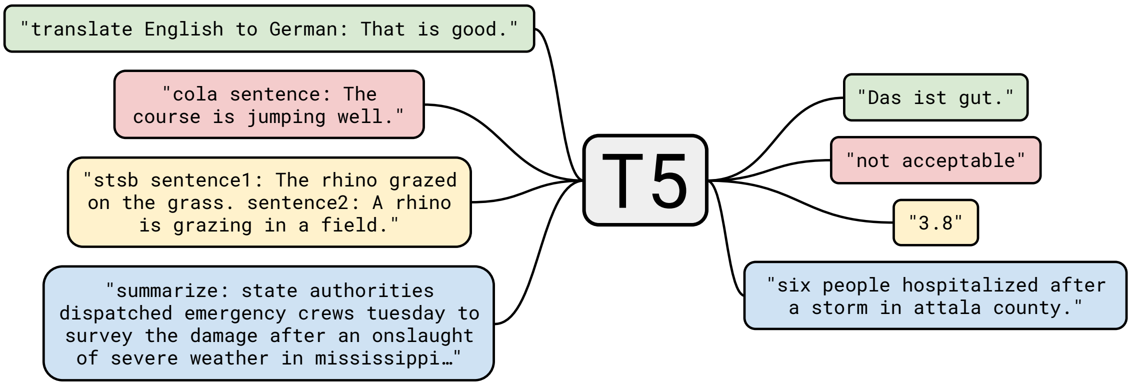

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

FIGURE 3.6: A diagram of text-to-text framework of [4]. "T5" refers to the model, which is dubbed as the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

be non-trivial since they might not fit on a single machine and require a great deal of computation. As a result, they use a combination of model and data parallelism and train models on "slices" of Cloud TPU Pods[1]. They leverage the Mesh TensorFlow library [141] for ease of implementation of both model parallelism and data parallelism [142].

3.3 Deep Learning for Audio

3.3.1 EnCodec

The EnCodec model is a convolutional encoder-decoder architecture designed for efficient audio compression and generation. It consists of three main components:

1. **Encoder (E)**: An encoder network that takes an audio signal as input and outputs a latent representation $\mathbf{z}$ using a series of convolutional layers and a sequence modeling component (LSTM).

2. **Quantization Layer (Q)**: A quantization layer that compresses the latent representation $\mathbf{z}$ into a discrete representation $\mathbf{z}_q$ using Residual Vector Quantization (RVQ). This enables efficient compression by quantizing the latent space.

3. **Decoder (G)**: A decoder network that reconstructs the time-domain audio signal $\hat{\mathbf{x}}$ from the compressed latent representation $\mathbf{z}_q$ using transposed convolutional layers.

[1] TPU pods are multi-rack ML supercomputers that contain 1,024 TPU v3 chips connected via a high-speed 2D mesh interconnect with supporting CPU host machines.

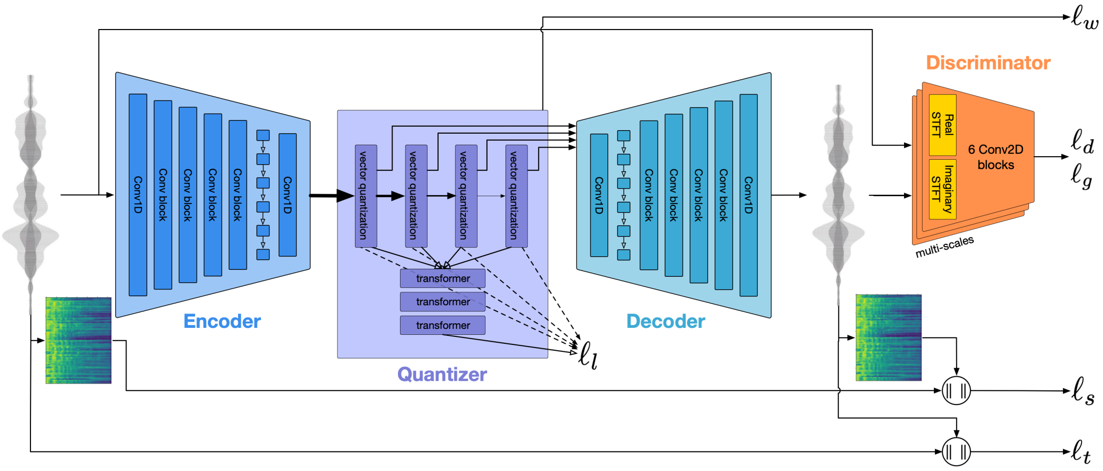


FIGURE 3.7: **EnCodec**: an encoder-decoder codec architecture which is trained with reconstruction (ℓ_f and ℓ_t) as well as adversarial losses (ℓ_g for the generator and ℓ_d for the discriminator). The residual vector quantization commitment loss (ℓ_w) applies only to the encoder. Optionally, we train a small Transformer language model for entropy coding over the quantized units with ℓ_l, which reduces bandwidth even further. [5]

The EnCodec model is trained end-to-end to minimize a reconstruction loss applied over both time and frequency domains, together with a perceptual loss in the form of discriminators operating at different resolutions. The encoder-decoder architecture is a simple streaming, convolutional-based model with sequential modeling components applied over the latent representation, both on the encoder and decoder sides. This modeling framework has shown great results in various audio-related tasks, such as source separation, enhancement, neural vocoders, audio codecs, and artificial bandwidth extension [143, 144, 145, 146, 52, 147, 148]. The RVQ layer quantizes the output of the encoder into a compressed representation by projecting the input vector onto the closest entry in a codebook of a given size. It refines this process by computing the residual after quantization and further quantizing it using additional codebooks. By selecting a variable number of residual steps during training, a single model can support multiple bandwidth targets. Additionally, a small Transformer-based language model is trained to estimate the probability distribution over the codebooks, enabling efficient entropy coding for compression and decompression faster than real-time on a single CPU core.

3.3.2 Residual Vector Quantizer

Neural Compression techniques have emerged as a new approach, employing neural networks to represent, compress, and reconstruct data, potentially achieving high compression rates with nearly zero perceptual information loss. In the audio domain, neural

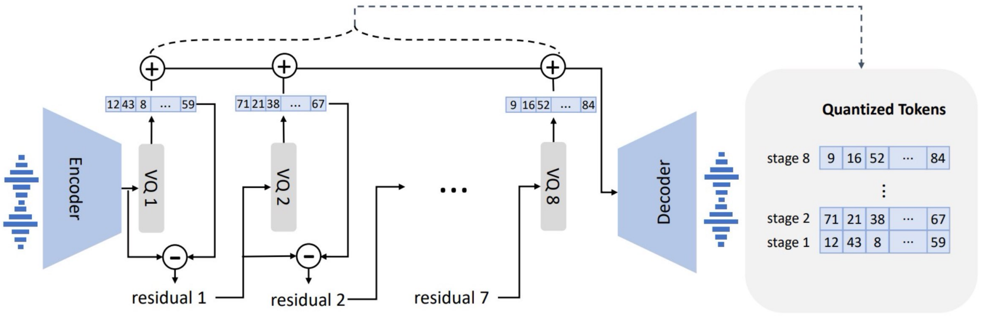


FIGURE 3.8: **RVQ** breaks down the quantization process across multiple layers, each handling the residual error from the preceding one. This allows the system to be scaled to operate on different bitrates (by scaling the number of layers).[6]

audio codecs based on Residual Vector Quantization have surpassed traditionally handcrafted pipelines, with state-of-the-art AI models like Google's SoundStream[149] and EnCodec[5] by Meta AI demonstrating proficiency in encoding audio signals across a broad spectrum of bitrates.

3.3.2.1 Neural Compression

Neural Compression aims to transform various data types, such as pixels (images), waveforms (audio), or frame sequences (video), into more compact representations, like vectors. Instead of recording every pixel value or waveform sample, Neural Compression learns to identify critical features or patterns in the data. These learned features are then used to reconstruct the data with high accuracy, analogous to the concept of autoencoders in deep learning.

3.3.2.2 Neural Audio Codecs

Neural audio codecs employ deep neural networks to translate recorded sound, a digital audio signal, into a given content format while maintaining the original qualities of the sound and reducing file size and bitrate. The goal is to achieve high compression rates while preserving perceptual quality.

3.3.2.3 Residual Vector Quantization (RVQ)

RVQ is a key component in state-of-the-art neural audio codecs, enabling high compression rates by quantizing high-dimensional vectors into lower-dimensional representations.

Basic Idea The central idea behind RVQ is to utilize a cascade of codebooks, each progressively approximating the residual error from the previous stage, rather than attempting to quantize high-dimensional vectors with a single, large codebook.

RVQ Algorithm The RVQ algorithm follows a general quantization process, as outlined in Algorithm 1 from the SoundStream paper [149]. The algorithm takes input data and a set of codebooks as input and outputs the indices of the quantized vectors. The process is as follows:

Algorithm 1: Residual Vector Quantization

```
Input  : $y = \mathrm{enc}(x)$ the output of the encoder, vector
         quantizers $Q_i$ for $i = 1 \ldots N_q$
Output: the quantized $\hat{y}$
$\hat{y} \leftarrow 0.0$
residual $\leftarrow y$
for $i \leftarrow 0$ to $N_q$ do
    $\hat{y} += Q_i(\text{residual})$
    residual $-= Q_i(\text{residual})$
end for
return $\hat{y}$
```

The algorithm iterates over a set of codebooks, where each codebook represents a quantization level. For each codebook, the algorithm finds the indices of the closest codewords (centroids) to the input data or residuals from the previous stage. These indices are stored, and the residuals are computed by subtracting the corresponding codewords from the input data or previous residuals. The process continues until all codebooks have been processed, and the final set of indices is returned.

Codebook Construction The codebooks used in RVQ can be constructed in various ways, such as using uniform hypercubes or more sophisticated partitioning schemes like k-means clustering. The choice of codebook construction method can impact the compression efficiency and reconstruction quality.

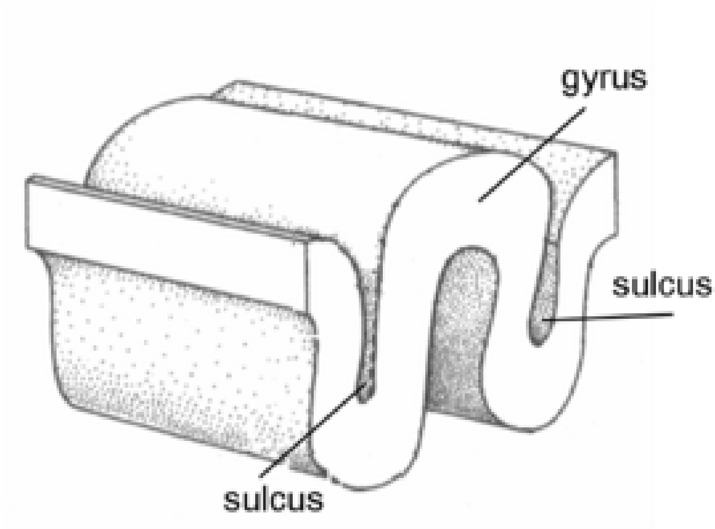


FIGURE 3.9: Cortex is made up of sulco-gyral structures.[7]

Computational Efficiency RVQ offers significant computational savings compared to traditional vector quantization (VQ) methods, especially in high-dimensional spaces. Instead of using a single, high-resolution codebook, RVQ employs a series of smaller codebooks, reducing the computational cost while maintaining good approximation accuracy.

3.3.2.4 Reconstruction and Decoding

After the quantization process, the compressed signal is passed to a decoder, which reconstructs the audio stream from the quantized indices and codebooks. The reconstructed audio is then compared to the original using a discriminator component, measuring losses such as discriminator/generator losses, waveform and mel-spectrogram losses, and commitment losses to ensure the output closely mirrors the initial input.

3.4 Brain Regions

The human brain is a highly complex and organized structure, consisting of various regions that work together to process and interpret sensory information, including auditory stimuli such as music. Of particular interest in the study of music perception are the regions of the cerebral cortex, the outermost layer of the brain responsible for higher-order cognitive functions [150].

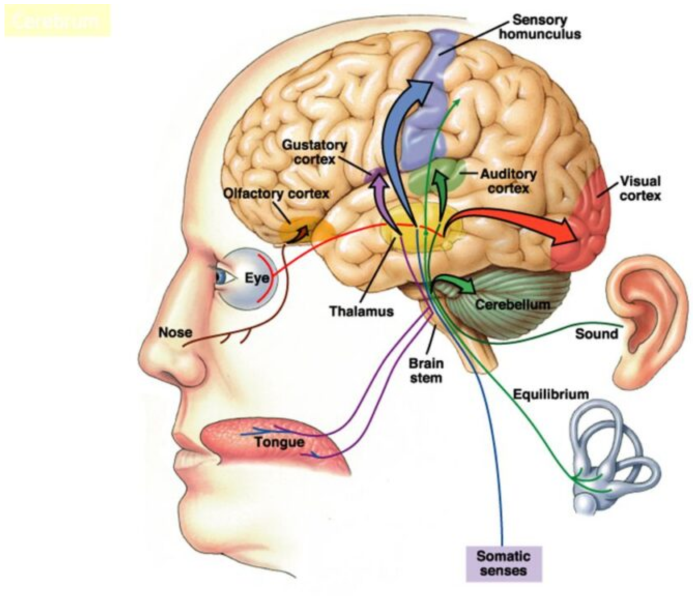


FIGURE 3.10: Pathways for Sensory Perception.[8]

3.4.1 Cerebral Cortex

The cerebral cortex is divided into four main lobes: frontal, parietal, temporal, and occipital. Each lobe is further subdivided into smaller regions, each with specific functions [151]. The surface of the cerebral cortex is characterized by a series of folds and ridges, known as sulci and gyri, respectively. Sulci are the grooves or fissures that separate the gyri, which are the bumps or ridges of the cortex. These structures increase the surface area of the cortex, allowing for greater processing power within the confined space of the skull [150].

3.4.1.1 Auditory Cortex

The auditory cortex, located in the temporal lobe, is crucial for the perception and processing of sound, including music [152]. It is divided into the primary auditory cortex (A1), secondary auditory cortex (A2), and higher-order auditory areas. A1, also known as the superior temporal gyrus (STG), is responsible for the initial processing of auditory information, such as pitch, frequency, and intensity discrimination [153]. The STG is further divided into several subregions, including the planum temporale (PT) and Heschl's gyrus (HG), which are particularly important for music perception [154]. A2 and higher-order auditory areas, such as the superior temporal sulcus (STS) and middle temporal gyrus (MTG), are involved in more complex auditory processing, such as the recognition

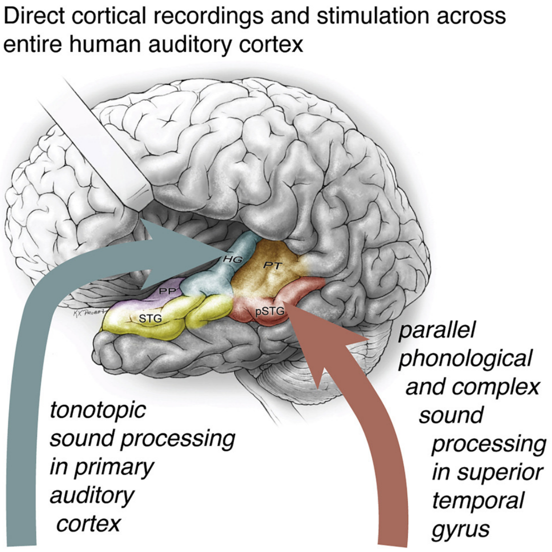

Direct cortical recordings and stimulation across entire human auditory cortex
HG
PT
PP
STG
pSTG
tonotopic sound processing in primary auditory cortex
parallel phonological and complex sound processing in superior temporal gyrus

FIGURE 3.11: **human auditory cortex**[9]

of melodies, harmonies, and rhythms [155]. These regions also contribute to the integration of auditory information with other sensory modalities, such as visual and motor information, which is essential for the holistic experience of music [152]. Studies using fMRI have shown that the auditory cortex is highly active during music listening tasks, with distinct patterns of activation observed for different musical features, such as pitch, timbre, and rhythm [155]. Moreover, the auditory cortex exhibits functional specialization, with certain subregions being more responsive to specific aspects of music.

3.4.1.2 Frontal Cortex

The frontal cortex, particularly the prefrontal cortex (PFC), is involved in higher-order cognitive functions, such as attention, working memory, and decision-making [156]. In the context of music perception, the PFC is thought to play a role in the cognitive processing of musical structure, such as the perception of key, harmony, and musical syntax [157, 158]. The inferior frontal gyrus (IFG), which includes Broca's area, is another important region in the frontal cortex for music perception. Although primarily associated with language processing, the IFG has also been implicated in the processing of musical syntax and the detection of musical violations [159, 160]. This suggests that there may be shared neural resources for the processing of structured sequences in both language and music.

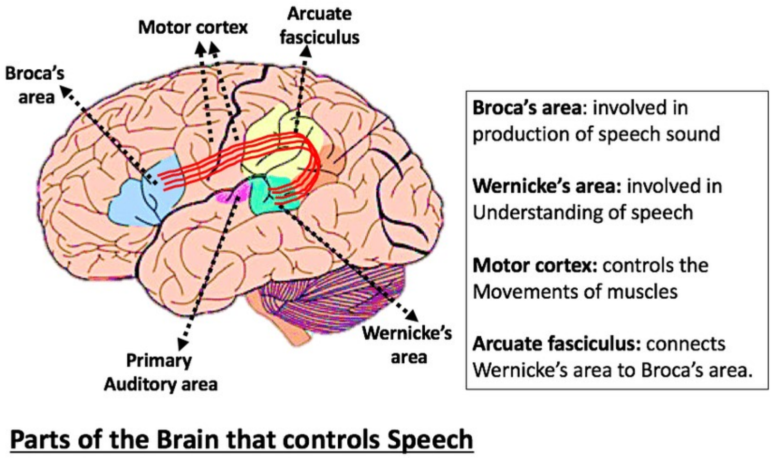

Motor cortex
Arcuate fasciculus
Broca's area
Broca's area: involved in production of speech sound
Wernicke's area: involved in Understanding of speech
Motor cortex: controls the Movements of muscles
Arcuate fasciculus: connects Wernicke's area to Broca's area.
Wernicke's area
Primary Auditory area
Parts of the Brain that controls Speech

FIGURE 3.12: **human Language Cortex**[10]

3.4.1.3 Parietal Cortex

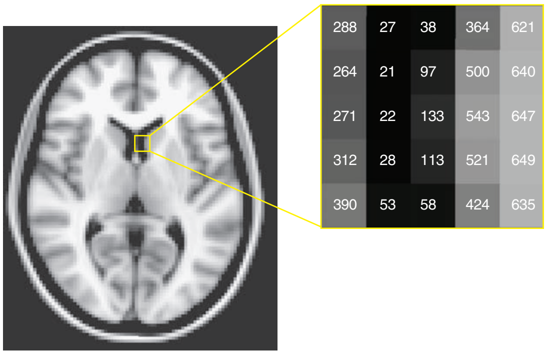
The parietal cortex, particularly the inferior parietal lobule (IPL), is involved in the processing of musical structure and the integration of auditory and motor information [152]. The IPL, which includes the supramarginal gyrus (SMG) and angular gyrus (AG), is thought to be important for the perception of musical meter and the temporal organization of sound [161]. The intraparietal sulcus (IPS) is another region in the parietal cortex that has been implicated in music perception. The IPS is involved in the processing of numerical and spatial information, and it has been shown to be active during tasks that involve the perception of musical intervals and the mental transformation of melodies [162].

3.5 fMRI Scans

Functional Magnetic Resonance Imaging (fMRI) is a powerful neuroimaging technique that allows researchers to measure brain activity by detecting changes in blood oxygenation levels. fMRI scans provide valuable insights into the functional organization of the brain and have become an essential tool in the field of cognitive neuroscience.

3.5.1 What is an Magnetic Resonance Image (MRI)?

We now give a brief description of what a Magnetic Resonance Image is. This can be important in order to understand some basic characteristics of the random variables we are going to use. It is worthwhile to point out that some preprocessing is necessary in this type of image in order to take raw data from the scanner and prepare them for statistical analysis. Some of the steps usually applied in the image preprocessing are motion correction, slice timing correction, spatial filtering, intensity normalization, and temporal filtering. In this case, an MRI can be viewed as a matrix of numbers that correspond to spatial locations. When we view an image, we do so by representing the numbers in the image in terms of grayscale values and each element in the image is called as a voxel, which is the three-dimensional analog to a pixel. See figure 3.13 for a visual description.



288	27	38	364	621
264	21	97	500	640
271	22	133	543	647
312	28	113	521	649
390	53	58	424	635

FIGURE 3.13: An image as a graphical representation of a matrix. The grayscale values in the image on the left correspond to numbers, shown for a specific set of voxels in the closeup section on the right ([**?**]).

3.5.2 What is an fMRI experiment?

An fMRI experiment involves an individual lying in an MRI scanner for a period of time, usually around 5 minutes, while their brain activity is measured. During this time, the scanner can acquire up to 100 low-resolution images of the brain (see the left panel of Figure 3.14). The individual may receive a sequence of stimuli according to a specific experimental design or may be asked to remain in a resting state without any external stimulation.

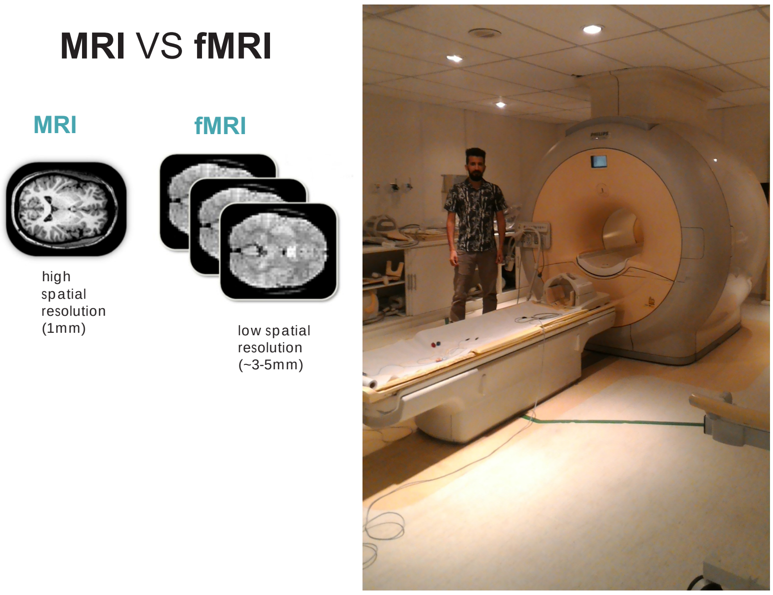

MRI VS fMRI
MRI
fMRI
high spatial resolution (1mm)
low spatial resolution (~3-5mm)

FIGURE 3.14: Left panel: Diferences between MRI and fMRI images. Right panel: The MRI scanner at the Institute of Radiology of the University of São Paulo.

The types of stimuli presented to the subject during an fMRI experiment depend on the research question being investigated and can include sensory, visual, and/or auditory stimuli. There are three main types of experimental designs used in fMRI experiments: block design, event-related design, and mixed design. The choice of design depends on the specific goals of the experiment and the trade-offs between statistical power and flexibility.

3.5.3 The BOLD signal and the hemodynamic response function (HRF)

fMRI measures brain activity indirectly by detecting changes in the blood-oxygenation-level-dependent (BOLD) signal. When neurons in a specific brain region become active, there is an increase in blood flow to that area, which delivers more oxygen than is needed to replenish the cells. This surplus of oxygenated blood leads to a change in the local magnetic properties, which can be detected by the MRI scanner. The time course of the BOLD signal in response to a brief stimulus is known as the hemodynamic response function (HRF) (see Figure 3.15). The HRF typically peaks around 4-6 seconds after the onset of neuronal activity and then gradually returns to baseline. By measuring the BOLD signal over time, fMRI can provide a picture of the neural activity in different brain regions during a task or in response to a stimulus.

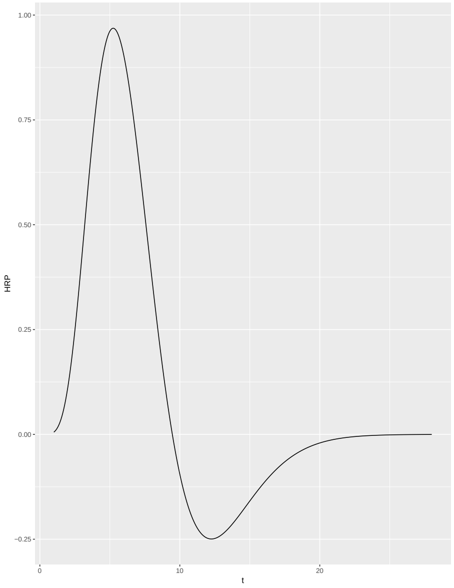
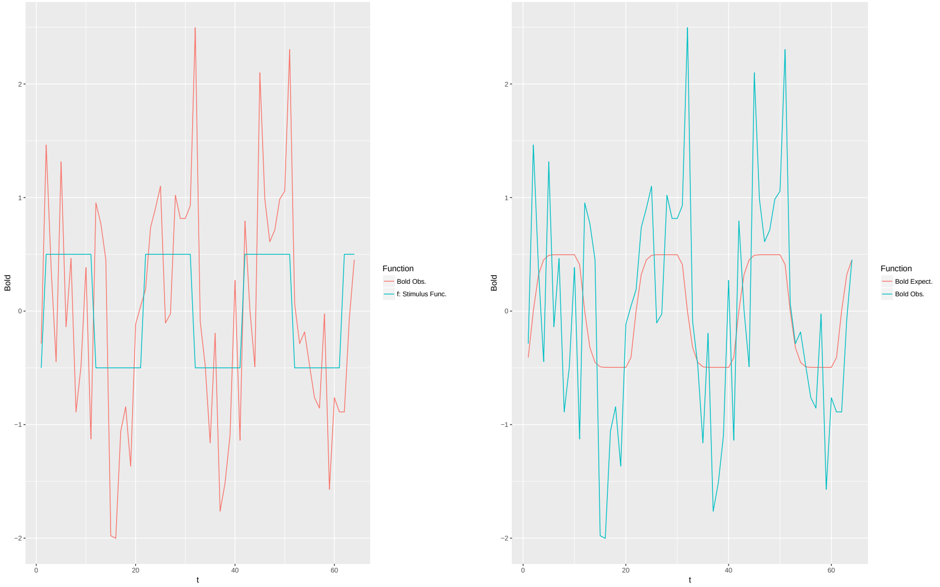


FIGURE 3.15: Hemodynamic Response Function:h.

$$x(t) = (h * f)(t) = \int h(\tau) f(t - \tau) d\tau \tag{3.18}$$

FIGURE 3.16: Left panel: Observed BOLD response (red line) and the stimulus time series (blue line). Right panel: Observed BOLD response (blue line) and the expected BOLD response (red line).



3.5.4 Mathematical Representation of fMRI data

The fMRI data $\mathbf{X}$ can be represented as a set of time series

$$\mathbf{X} = \{\boldsymbol{x}_{\boldsymbol{v}} | \boldsymbol{v}^{(0)} \in [0, W), \boldsymbol{v}^{(1)} \in [0, H), \boldsymbol{v}^{(2)} \in [0, D)\}. \tag{3.19}$$

Here $\boldsymbol{v} = [\boldsymbol{v}^{(0)}, \boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}]$ represents the 3D spatial coordinates, and $\boldsymbol{x}_{\boldsymbol{v}} \in \mathbb{R}^T$ represents the fMRI signal time series at the corresponding spatial location

$$\boldsymbol{x}_{\boldsymbol{v}} = [\boldsymbol{x}_{\boldsymbol{v}}^{(0)}, \boldsymbol{x}_{\boldsymbol{v}}^{(1)}, \cdots, \boldsymbol{x}_{\boldsymbol{v}}^{(T-1)}], \tag{3.20}$$

with W, H, D, T respectively denoting the width, height, depth, and length of the time series in the fMRI data.

The principle of localization in brain function organization suggests that brain functions are carried out in a set of brain regions [163]. This implies that the brain space can be divided into several brain regions based on function, and there are similarities in neuronal activation within each brain region. Therefore, in fMRI data, the set of time series for each brain region can be represented as

$$\mathbf{A}_i = \{\boldsymbol{x}_{\boldsymbol{v}} | \boldsymbol{v} \in \mathbf{V}_i\}, i \in [0, N), \tag{3.21}$$

in which $\mathbf{V}_i$ represents the set of three-dimensional coordinates for the brain region labeled as i, and N represents the number of brain region labels. Therefore, we have

$$\mathbf{X} = \bigcup_{i=0}^{N-1} \mathbf{A}_i, \tag{3.22}$$

implying that V can be represented as the union of $\mathbf{A}_i$.

Chapter 4

Methodology

4.1 Task Overview

Let $\mathcal{G}$ be a conditional music generation model that can be conditioned on inputs from different modalities defined in set M. The model would have modality-specific encoders E_m that map input in modality $m \in M$ to a common prior space of $\mathcal{G}$ to guide the generation of music. Thus for a conditional input c_m of modality $m \in M$, music g_m is generated as follows using the modality encoder E_m:

$$g_m = \mathcal{G}(E_m(c_m)) \tag{4.1}$$

Let $B \in \mathbb{R}^{s \times v}$ denote the fMRI response tensor for a subject listening to a music sample $\mathcal{K}$, where s is the number of scans taken and v is the number of voxel intensities. Each voxel intensity corresponds to the BOLD response at a particular location of the subject's brain during the scan. For music sample $\mathcal{K}$, let k be one of its modality representations and $T \in \mathbb{R}^{r \times d}$ be the computed embedding of k using encoder E_k, where r is the number of embeddings computed for the length of the sample and d being the dimension of embeddings.

Following the Map method [164], the task is to learn a mapping $\Phi : B \rightarrow T$, i.e. the brain scans are mapped to prior embedding space of $\mathcal{G}$. Using the learned mapping, the listened music $\mathcal{K}$ can be reconstructed into $\mathcal{K}_r$:

$$\mathcal{K}_r = \mathcal{G}(\Phi(B)) \tag{4.2}$$

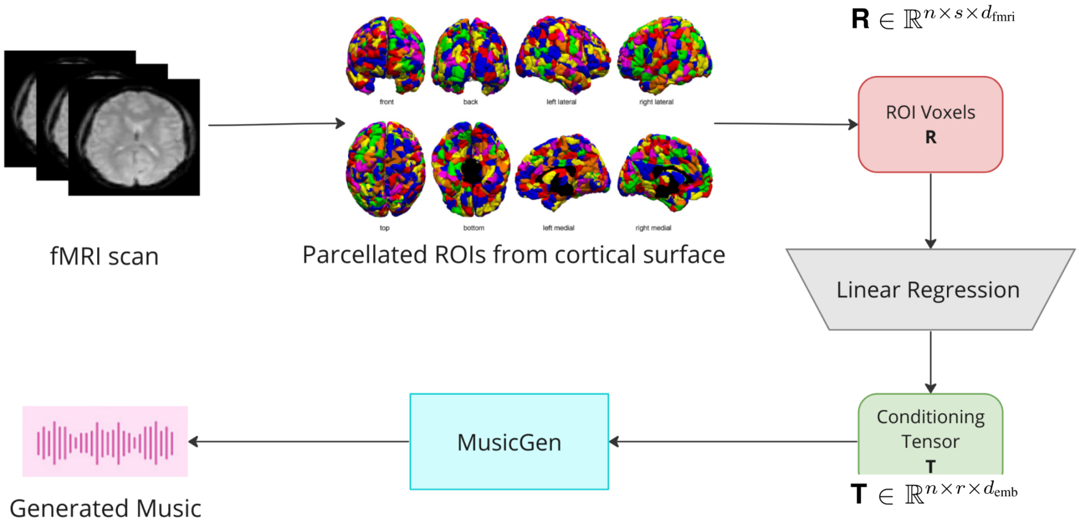

fMRI scan
front
back
left lateral
right lateral
top
bottom
left medial
right medial
Parcellated ROIs from cortical surface
$\mathbf{R} \in \mathbb{R}^{n \times s \times d_{\text{fmri}}}$
ROI Voxels
R
Linear Regression
Conditioning Tensor
T
$\mathbf{T} \in \mathbb{R}^{n \times r \times d_{\text{emb}}}$
MusicGen
Generated Music

Figure 4.1: Proposed Pipeline for Music Generation from Brain Scans.

The mapping Φ is parameterized by a function f_θ with parameters θ, and is learned from a training set $\mathcal{D} = \{B_i, E_k(\mathcal{K}_i)\}$, where B_i and $E_k(\mathcal{K}_i)$ are the brain scan sequences and corresponding prior embeddings computed from a specific modality representation of music sample $\mathcal{K}_i$. The learned mapping $\Phi = f_\theta$ can then be applied to unseen brain scan sequences B_s to predict the corresponding prior embeddings T_s, enabling the generation of reconstructed music stimuli $\mathcal{K}'_s$ via the generative model $\mathcal{G}$.

Practically, we implement MusicGen [11] as the music generation model $\mathcal{G}$. We experiment with audio waveforms, text, and chromagrams as modality representations of music samples and their corresponding encoders EnCodec [5], T5 [4], and quantizer respectively, to learn the mapping function Φ modelled as an L-2 regularized linear regression. The raw fMRI scans of 5 subjects listening to 540 music samples are processed to extract voxel-time-series on the cortical surfaces in particular 148 regions of interest based on the Destriuex Atlas [7] defined as set $\mathcal{C}$. An ensemble of Φ_r, $\forall r \in \mathcal{C}$, models are learned for each ROI independently, picking the top models based on the correlation score and averaging their predictions.

Figure 4.1 shows the proposed pipeline of our methodology. We process raw fMRI scans to parcelllate the cortical surface to identify regions of interest as per the Destrieux atlas. From these ROI voxels or voxel time series, the conditioning tensor is predicted using a trained L2-regularized linear regression ensemble model. Using the condition tensor, MusicGen generates the listened music, thus achieving our goal.

The preceding formulation encapsulates the principles involved in reconstructing music from brain activity, without diving into specific implementation details, which shall be discussed in subsequent sections.

4.2 Extracting Cortical Voxel Time-series

In the field of functional neuroimaging, the accurate extraction of voxel time series from specific cortical regions is crucial for understanding how different brain areas respond to experimental stimuli or cognitive tasks. This section outlines the methodology employed to obtain cortical voxel time series from functional Magnetic Resonance Imaging (fMRI) data, leveraging the high-resolution anatomical information provided by structural T1-weighted MRI scans. The analysis pipeline begins with the acquisition of two distinct types of MRI data for each subject: (1) a high-resolution T1-weighted structural scan, which provides excellent contrast between gray matter, white matter, and cerebrospinal fluid (CSF), and (2) a series of T2*-weighted functional scans, which capture the Blood Oxygenation Level Dependent (BOLD) signal changes associated with neural activity over time. While the structural scan offers detailed anatomical information, the functional scans lack sufficient spatial resolution and contrast to delineate cortical structures accurately.

To overcome this limitation, we employ a surface-based analysis approach, which involves reconstructing the cortical surface from the structural scan and subsequently mapping the functional data onto this reconstructed surface. This process enables the extraction of voxel time series from specific cortical regions defined by anatomical atlases, facilitating region-of-interest (ROI) analyses and structure-function investigations.

The following subsections detail the key steps involved in this process, beginning with the reconstruction of the cortical surface from the structural scan and its parcellation using the Destrieux Atlas [7]. Subsequently, we describe the registration of the functional scans to the structural scan space, allowing the mapping of the parcellated cortical regions onto the functional data. This integration of anatomical and functional information ultimately enables the extraction of cortical voxel time series from the fMRI data, corresponding to the predefined ROIs specified by the Destrieux Atlas.

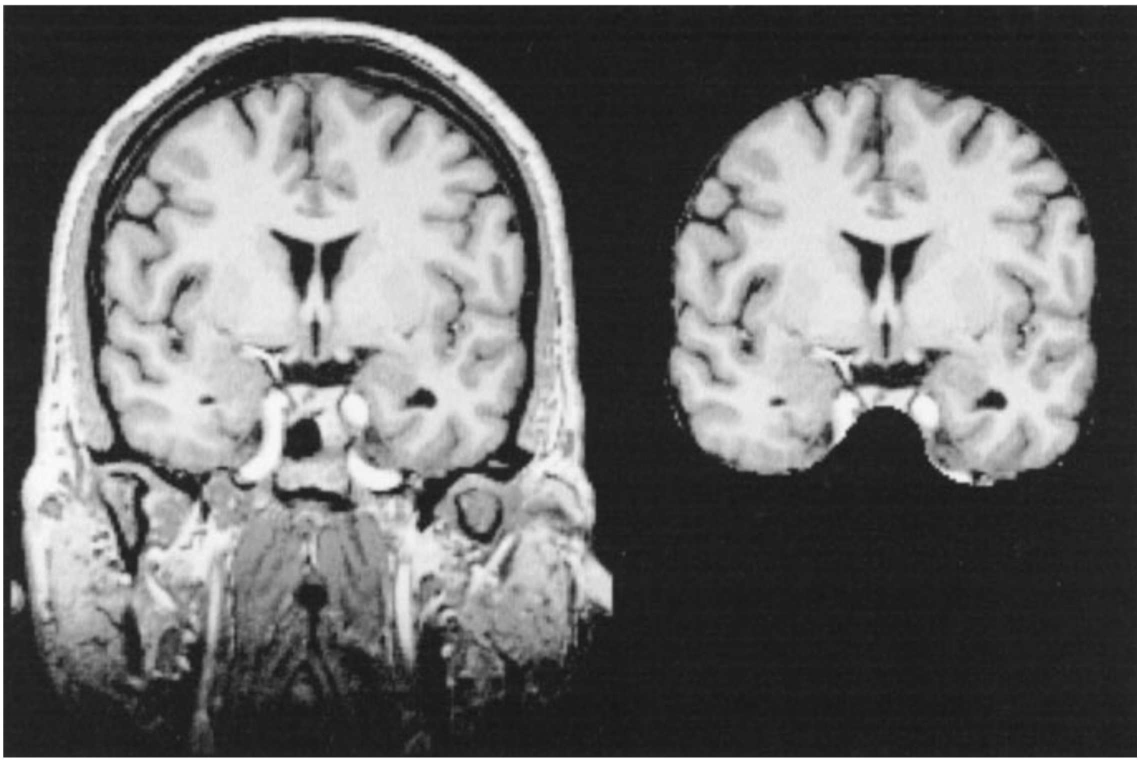


Figure 4.2: Intensity normalized (left) and skull-stripped images (right)

4.2.1 Reconstruction of Cortical Surface

The accurate reconstruction of the cortical surface from structural MRI data is a crucial step in enabling surface-based analysis of brain structure and function. Let $\mathcal{I} \in \mathbb{R}^{W \times H \times D}$ denote the T1-weighted MRI volume of a subject, where W, H, and D represent the width, height, and depth of the volume, respectively.

To correct for intensity inhomogeneities arising from magnetic field distortions and RF field non-uniformities, a non-parametric non-uniform intensity normalization (N3) technique [165] is applied to $\mathcal{I}$, resulting in a corrected volume $\mathcal{I}_c$. Subsequently, non-brain tissues such as the skull and dura are removed from $\mathcal{I}_c$ using a deformable model approach [166], yielding a skull-stripped volume $\mathcal{I}_s$ as shown in figure 4.2.

The intensity-normalized, skull-stripped volume $\mathcal{I}_s$ is then segmented into white matter, gray matter, and non-brain tissues using a custom segmentation algorithm [166]. This method exploits the locally planar structure of the gray/white matter interface by detecting the plane of least variance $\mathcal{P}_v(\mathbf{x})$ at each voxel $\mathbf{x} \in \mathbb{R}^3$ and using intensity information within $\mathcal{P}_v(\mathbf{x})$ to guide the segmentation decision. Geometric constraints are also incorporated based on the knowledge that white matter has higher intensity than gray matter

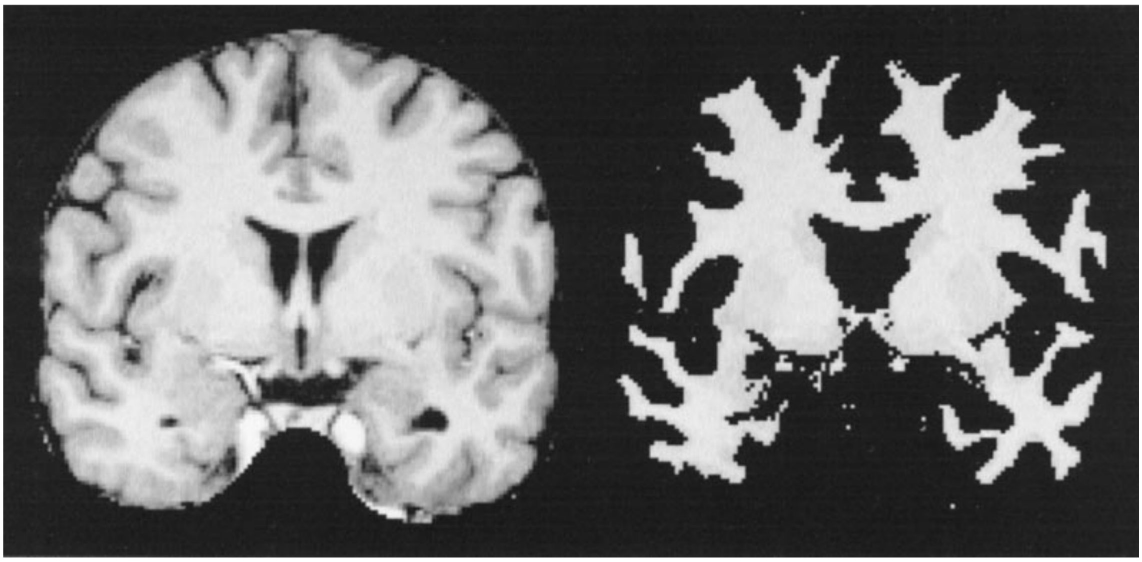


FIGURE 4.3: Skull-stripped (left) and white matter-labeled images (right)

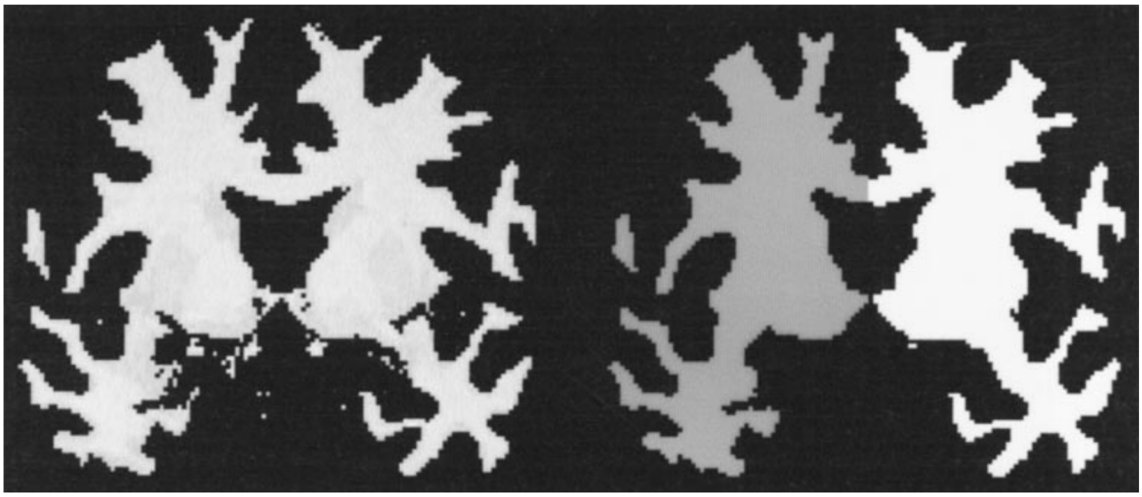


FIGURE 4.4: Result of calculating connected components (right) for white matter-labeled volume (left). Connected right and left hemisphere voxels are labeled dark and light, respectively.

in $\mathcal{I}_s$. The segmentation process results in a binary volume $\mathcal{S} \in 0, 1^{W \times H \times D}$ as shown in figure 4.3, where $\mathcal{S}(\mathbf{x}) = 1$ if $\mathbf{x}$ is classified as white matter, and $\mathcal{S}(\mathbf{x}) = 0$ otherwise.

Two cutting planes, $\mathcal{C}_1$ and $\mathcal{C}_2$, are computed to separate the cerebral hemispheres as well as disconnect subcortical regions from the cortical components [166]. A connected components analysis is then performed on $\mathcal{S}$ to obtain two hemispheric volumes $\mathcal{H}_L$ and $\mathcal{H}_R$, representing the left and right hemispheres, respectively as shown in figure 4.4. Any interior holes in $\mathcal{H}_L$ and $\mathcal{H}_R$ are filled to create topologically closed surfaces.

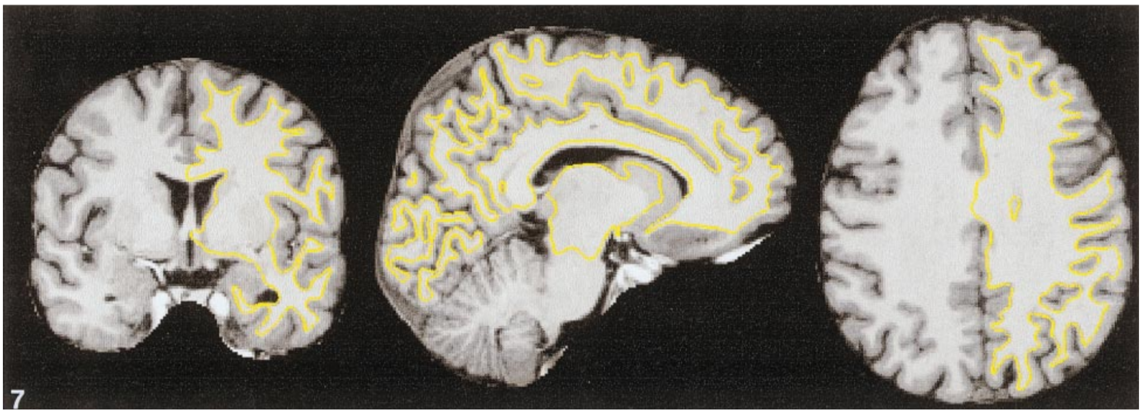


FIGURE 4.5: Intersection of the tessellated white matter surface with the skull-stripped MRI volume.

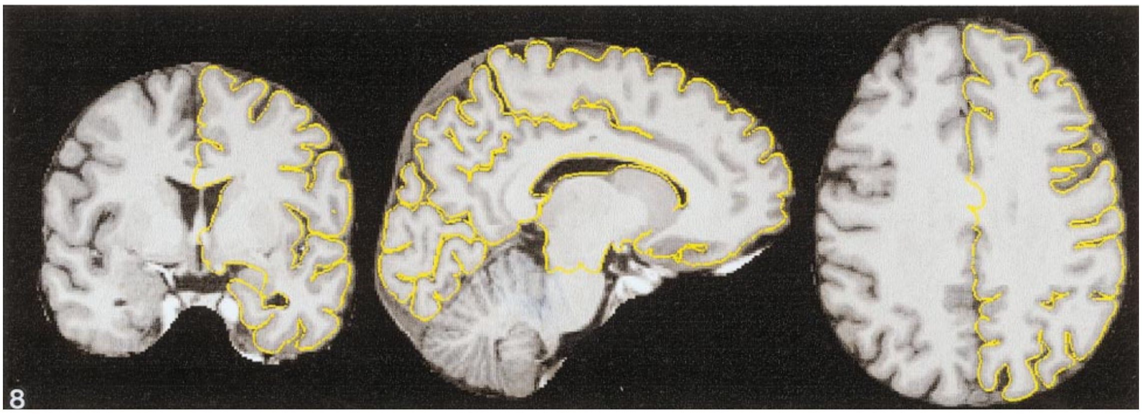


FIGURE 4.6: Intersection of the tessellated pial surface with the skull-stripped MRI volume

The hemispheric white matter volumes $\mathcal{H}_L$ and $\mathcal{H}_R$ are covered with triangular tessellations $\mathcal{T}_L$ and $\mathcal{T}_R$, respectively, by representing each face between a white matter voxel and an adjacent non-white matter voxel using two triangles [166]. These initial tessellated surfaces are deformed using a deformable surface algorithm to obtain accurate reconstructions $\mathcal{S}_L$ and $\mathcal{S}_R$ of the gray/white matter interface for the left and right hemispheres, respectively as shown in figure 4.5. A similar procedure is employed to reconstruct the pial surfaces $\mathcal{P}_L$ and $\mathcal{P}_R$ by deforming $\mathcal{S}_L$ and $\mathcal{S}_R$ outwards towards the gray matter/CSF boundary as shown in figure 4.6, guided by the intensity values in $\mathcal{I}_s$.

The resulting surface reconstructions $\mathcal{S}_L$, $\mathcal{S}_R$, $\mathcal{P}_L$, and $\mathcal{P}_R$ typically contain topological defects, primarily in subcortical regions, that prevent them from being accurately unfolded or inflated. These defects are manually edited by adding control points to the segmented volume $\mathcal{S}$ and recomputing the tessellations $\mathcal{T}_L$ and $\mathcal{T}_R$, guided by the surface inflation procedure [166, 167].

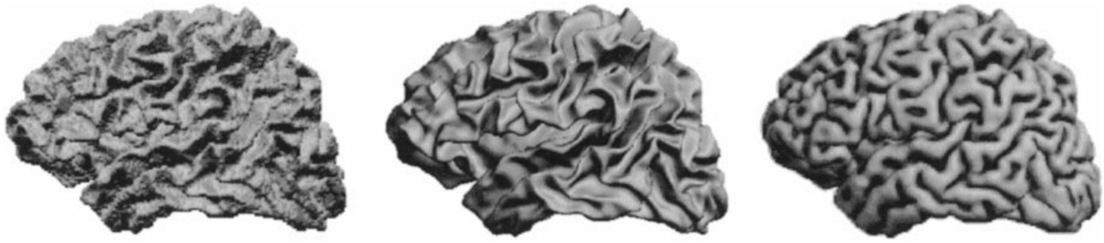


Figure 4.7: Original (left), gray/white boundary (middle), and pial surface (right) reconstructions of a left hemisphere.

The corrected cortical surface reconstructions in figure 4.7 $\mathcal{S}_L$, $\mathcal{S}_R$, $\mathcal{P}_L$, and $\mathcal{P}_R$ enable surface-based analysis, visualization, and cross-subject averaging, as described in subsequent sections.

4.2.2 Cortical Surface Parcellation Using Atlases

The reconstructed cortical surface is parcellated into anatomically distinct regions using surface-based atlases. This parcellation is crucial for performing region-of-interest (ROI) analyses and investigating structure-function relationships in the human brain. We employ the Destrieux Atlas [7], a surface-based parcellation scheme that subdivides the cortical surface into 74 regions of interest (ROIs) (refer table 4.1) in both hemispheres based on sulcal patterns and anatomical landmarks as shown in figure 4.8. Thus we have a total of 148 ROIs in the whole brain.

Let $\mathcal{S}$ be the reconstructed cortical surface mesh, represented as a set of vertices $\mathcal{V}$ and triangular faces $\mathcal{F}$. The parcellation process involves establishing a mapping $\phi : \mathcal{S} \to \mathcal{A}$, where $\mathcal{A}$ is the atlas space defined by the Destrieux Atlas. This mapping is achieved through a non-rigid surface registration technique that aligns the individual subject's cortical surface with the atlas template. Specifically, we employ a spherical harmonic-based surface registration algorithm [167] that minimizes the mean squared difference between the folding patterns of the subject's cortical surface and the atlas template. The folding patterns are quantified using the average convexity measure $C(\mathbf{v})$ defined at each vertex $\mathbf{v} \in \mathcal{V}$ of the surface mesh:

$$C(\mathbf{v}) = \int_0^T \frac{\partial J_s}{\partial \mathbf{v}} \cdot \mathbf{n}(\mathbf{v}, t), dt \tag{4.3}$$

Here, J_s is the energy functional used for surface inflation [167], $\mathbf{n}(\mathbf{v}, t)$ is the unit normal vector at vertex $\mathbf{v}$ and time t, and T is the total number of time steps in the inflation

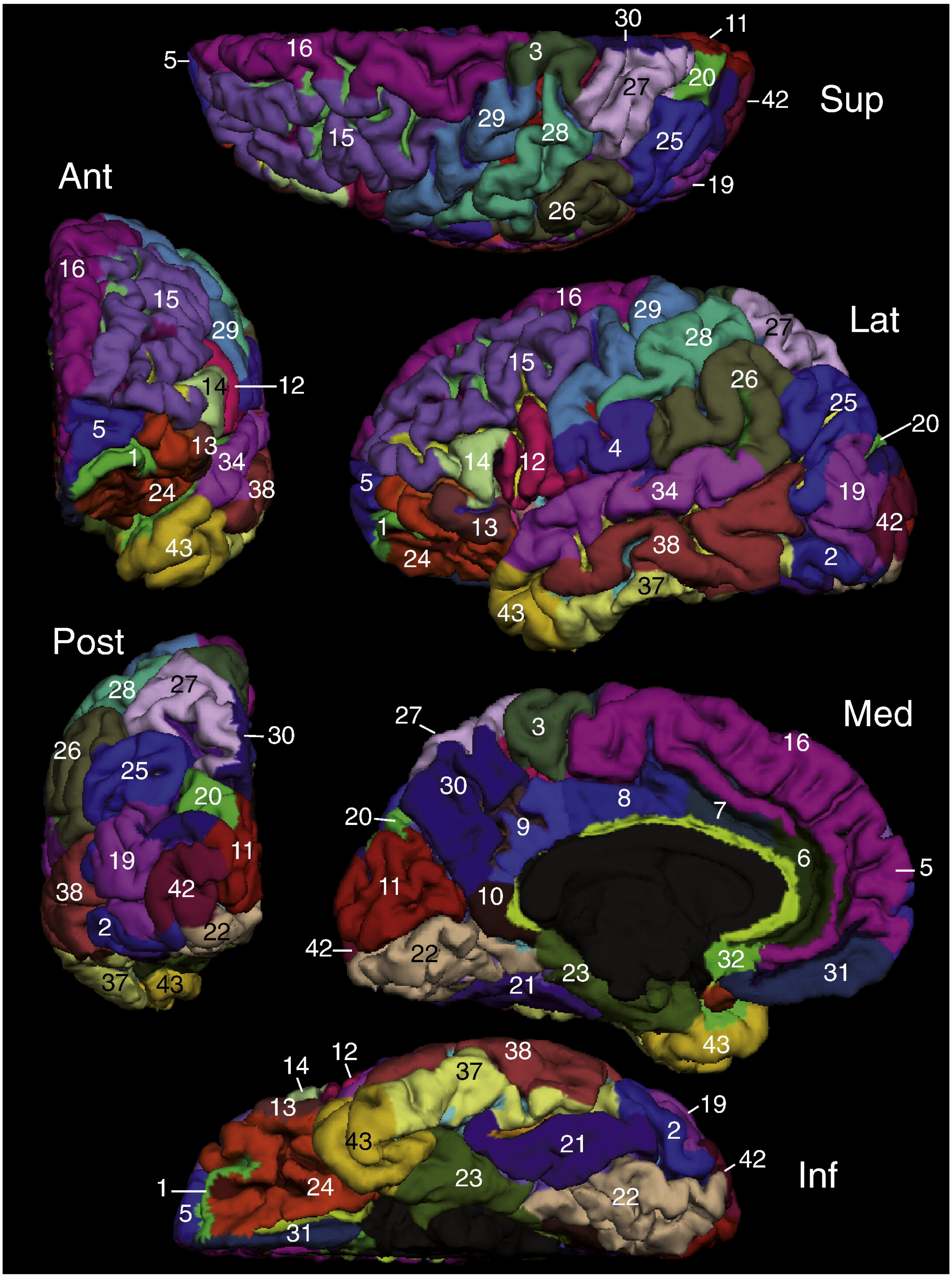


FIGURE 4.8: Pial view of the Destrieux Atlas Parcellation Regions

process. The average convexity $C(\mathbf{v})$ captures the large-scale folding patterns of the cortical surface while being relatively insensitive to small folds and noise. The registration algorithm finds the optimal mapping ϕ that maximizes the correlation between the average convexity maps of the subject's cortical surface and the atlas template. This mapping is then used to transfer the anatomical labels from the atlas space $\mathcal{A}$ onto the subject's cortical surface $\mathcal{S}$, resulting in a parcellated surface with 148 distinct ROIs based on the Destrieux Atlas.

The parcellated cortical surface serves as the basis for subsequent region-of-interest analyses, enabling the investigation of structural and functional properties within anatomically defined regions. Furthermore, the surface-based parcellation facilitates group-level analyses by establishing a common coordinate system across subjects, allowing for accurate inter-subject averaging and comparison of cortical measures.

4.2.3 Registration of Functional Scans to Structural Scans

The accurate registration of functional MRI scans to high-resolution structural scans is a crucial step in enabling the mapping of the parcellated cortical surface onto the functional data, thereby allowing the extraction of cortical voxel time-series for subsequent analysis. This registration process aims to establish a spatial correspondence between the functional and structural images, accounting for potential misalignments arising from subject motion, image distortions, and differences in acquisition parameters.

The effective approach for this registration task is the Boundary-Based Registration (BBR) method [168]. BBR treats the structural and functional images asymmetrically, leveraging the high-quality anatomical contrast of the structural scan to extract surfaces that separate brain tissues and structures. The functional image, which may exhibit lower resolution or intensity inhomogeneities, is then aligned to the structural reference by maximizing the intensity gradient across the tissue boundaries delineated by the extracted surfaces.

The registration process involves the following key steps as shown in figure 4.9:

1. Assume the functional and structural images are in roughly the same location. If not, align the outlines of the images.

2. Exploit the different contrast weightings of the anatomical and functional images, a property known as mutual information. Areas that appear dark in the structural image (e.g., cerebrospinal fluid) will appear bright in the functional image, and vice

Table 4.1: List of anatomical parcellations in Destrieux Atlas [7]

Index	Short name	Long name (TA nomenclature is bold typed)
1	G_and_S_frontomargin	Fronto-marginal gyrus (of Wernicke) and sulcus
2	G_and_S_occipital_inf	Inferior occipital gyrus (O3) and sulcus
3	G_and_S_paracentral	Paracentral lobule and sulcus
4	G_and_S_subcentral	Subcentral gyrus (central operculum) and sulci
5	G_and_S_transv_frontopol	Transverse frontopolar gyri and sulci
6	G_and_S_cingul-Ant	Anterior part of the cingulate gyrus and sulcus (ACC)
7	G_and_S_cingul-Mid-Ant	Middle-anterior part of the cingulate gyrus and sulcus (aMCC)
8	G_and_S_cingul-Mid-Post	Middle-posterior part of the cingulate gyrus and sulcus (pMCC)
9	G_cingul-Post-dorsal	Posterior-dorsal part of the cingulate gyrus (dPCC)
10	G_cingul-Post-ventral	Posterior-ventral part of the cingulate gyrus (vPCC, isthmus of the cingulate gyrus)
11	G_cuneus	Cuneus (O6)
12	G_front_inf-Opercular	Opercular part of the inferior frontal gyrus
13	G_front_inf-Orbital	Orbital part of the inferior frontal gyrus
14	G_front_inf-Triangul	Triangular part of the inferior frontal gyrus
15	G_front_middle	Middle frontal gyrus (F2)
16	G_front_sup	Superior frontal gyrus (F1)
17	G_Ins_lg_and_S_cent_ins	Long insular gyrus and central sulcus of the insula
18	G_insular_short	Short insular gyri
19	G_occipital_middle	Middle occipital gyrus (O2, lateral occipital gyrus)
20	G_occipital_sup	Superior occipital gyrus (O1)
21	G_oc-temp_lat-fusifor	Lateral occipito-temporal gyrus (fusiform gyrus, O4-T4)
22	G_oc-temp_med-Lingual	Lingual gyrus, lingual part of the medial occipito-temporal gyrus, (O5)
23	G_oc-temp_med-Parahip	Parahippocampal gyrus, parahippocampal part of the medial occipito-temporal gyrus, (T5)
24	G_orbital	Orbital gyri
25	G_pariet_inf-Angular	Angular gyrus
26	G_pariet_inf-Supramar	Supramarginal gyrus
27	G_parietal_sup	Superior parietal lobule (lateral part of P1)
28	G_postcentral	Postcentral gyrus
29	G_precentral	Precentral gyrus
30	G_precuneus	Precuneus (medial part of P1)
31	G_rectus	Straight gyrus, Gyrus rectus
32	G_subcallosal	Subcallosal area, subcallosal gyrus
33	G_temp_sup-G_T_transv	Anterior transverse temporal gyrus (of Heschl)
34	G_temp_sup-Lateral	Lateral aspect of the superior temporal gyrus
35	G_temp_sup-Plan_polar	Planum polare of the superior temporal gyrus
36	G_temp_sup-Plan_tempo	Planum temporale or temporal plane of the superior temporal gyrus
37	G_temporal_inf	Inferior temporal gyrus (T3)
38	G_temporal_middle	Middle temporal gyrus (T2)
39	Lat_Fis-ant-Horizont	Horizontal ramus of the anterior segment of the lateral sulcus (or fissure)
40	Lat_Fis-ant-Vertical	Vertical ramus of the anterior segment of the lateral sulcus (or fissure)
41	Lat_Fis-post	Posterior ramus (or segment) of the lateral sulcus (or fissure)
42	Pole_occipital	Occipital pole
43	Pole_temporal	Temporal pole
44	S_calcarine	Calcarine sulcus
45	S_central	Central sulcus (Rolando's fissure)
46	S_cingul-Marginalis	Marginal branch (or part) of the cingulate sulcus
47	S_circular_insula_ant	Anterior segment of the circular sulcus of the insula
48	S_circular_insula_inf	Inferior segment of the circular sulcus of the insula
49	S_circular_insula_sup	Superior segment of the circular sulcus of the insula
50	S_collat_transv_ant	Anterior transverse collateral sulcus
51	S_collat_transv_post	Posterior transverse collateral sulcus
52	S_front_inf	Inferior frontal sulcus
53	S_front_middle	Middle frontal sulcus
54	S_front_sup	Superior frontal sulcus
55	S_interm_prim-Jensen	Sulcus intermedius primus (of Jensen)
56	S_intrapariet_and_P_trans	Intraparietal sulcus (interparietal sulcus) and transverse parietal sulci
57	S_oc_middle_and_Lunatus	Middle occipital sulcus and lunatus sulcus
58	S_oc_sup_and_transversal	Superior occipital sulcus and transverse occipital sulcus
59	S_occipital_ant	Anterior occipital sulcus and preoccipital notch (temporo-occipital incisure)
60	S_oc-temp_lat	Lateral occipito-temporal sulcus
61	S_oc-temp_med_and_Lingual	Medial occipito-temporal sulcus (collateral sulcus) and lingual sulcus
62	S_orbital_lateral	Lateral orbital sulcus
63	S_orbital_med-olfact	Medial orbital sulcus (olfactory sulcus)
64	S_orbital-H_Shaped	Orbital sulci (H-shaped sulci)
65	S_parieto_occipital	Parieto-occipital sulcus (or fissure)
66	S_pericallosal	Pericallosal sulcus (S of corpus callosum)
67	S_postcentral	Postcentral sulcus
68	S_precentral-inf-part	Inferior part of the precentral sulcus
69	S_precentral-sup-part	Superior part of the precentral sulcus
70	S_suborbital	Suborbital sulcus (sulcus rostrales, supraorbital sulcus)
71	S_subparietal	Subparietal sulcus
72	S_temporal_inf	Inferior temporal sulcus
73	S_temporal_sup	Superior temporal sulcus (parallel sulcus)
74	S_temporal_transverse	Transverse temporal sulcus

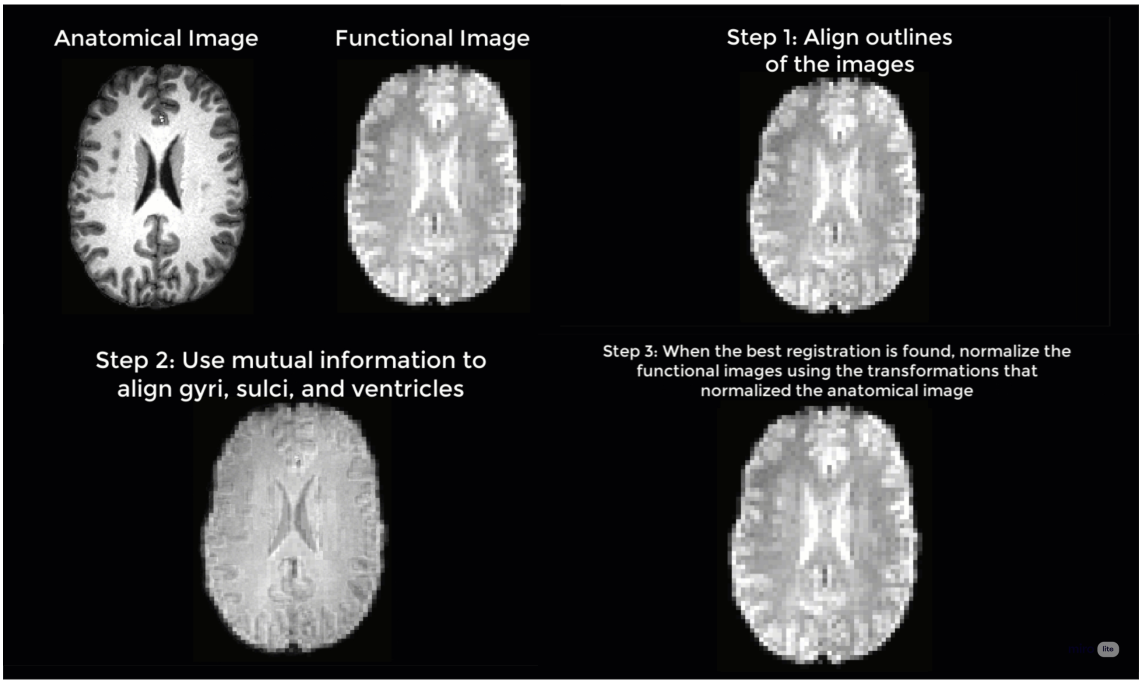

Anatomical Image
Functional Image
Step 1: Align outlines of the images
Step 2: Use mutual information to align gyri, sulci, and ventricles
Step 3: When the best registration is found, normalize the functional images using the transformations that normalized the anatomical image
lite

Figure 4.9: Boundary Based Registration Pipeline

versa. The registration algorithm moves the images around to test different overlays, matching the bright voxels on one image with the dark voxels of the other image, and dark with bright, until it finds an optimal match that cannot be improved further.

3. Once the best match has been found, apply the same transformations used to warp the structural image to the template space to the functional images.

After this registration step, the functional images are now aligned with the structural scan space, enabling the mapping of the parcellated cortical regions obtained from the Destrieux Atlas onto the functional data. Consequently, the voxel time series corresponding to each of the 148 ROIs defined by the Destrieux Atlas can be extracted from the registered functional scans.

4.3 Conditional Music Generation Model: MusicGen

In this section, we describe the MusicGen model [11] for conditional music generation, which is capable of generating music given textual descriptions, audio waveforms, or chromagrams as input. We discuss the model architecture, conditioning on different modalities, and training and evaluation procedures.

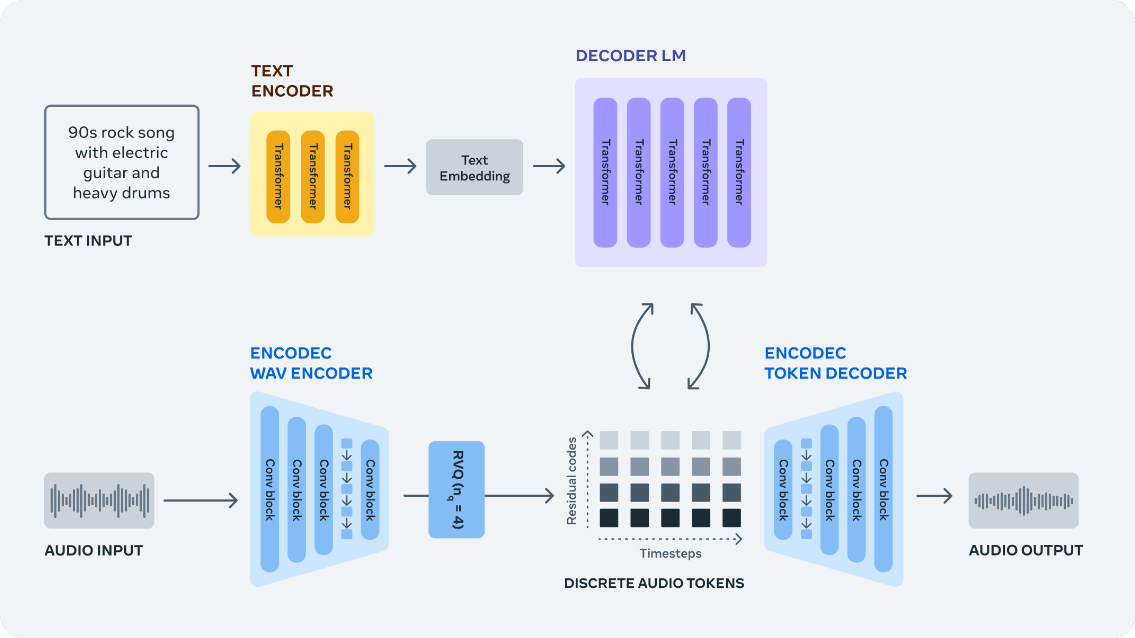


FIGURE 4.10: **MusicGen** architechture by Meta. [11]

4.3.1 Model Architecture and Music Generation

Music generation is a challenging task that involves creating original and coherent musical pieces. Recent advances in deep learning have led to the development of models that can generate music by modeling it as a sequence of tokens using transformers. MusicGen is one such model that uses an autoregressive transformer architecture to generate music tokens, conditioned on the input modality encodings.

The MusicGen model is trained on a large dataset of music samples, conditioned on different modalities. During training, the model learns to generate new music samples that are similar to the input samples, while also being conditioned on the input modality encodings. The model uses causal self-attention to ensure that the generated tokens are only conditioned on the previous tokens in the sequence. The model also uses cross-attention to condition the generation on the input modality encodings.

The MusicGen model is based on an autoregressive transformer [2] architecture, which generates music one token at a time. The model takes as input a sequence of tokens, where each token represents a quantized audio frame or a text or chromagram token. The model then predicts the next token in the sequence, conditioned on the previous tokens.The key idea is the use of codebook interleaving patterns to efficiently model the parallel streams from the RVQ quantization. Each pattern $P = (P_0, P_1, \ldots, P_S)$ defines

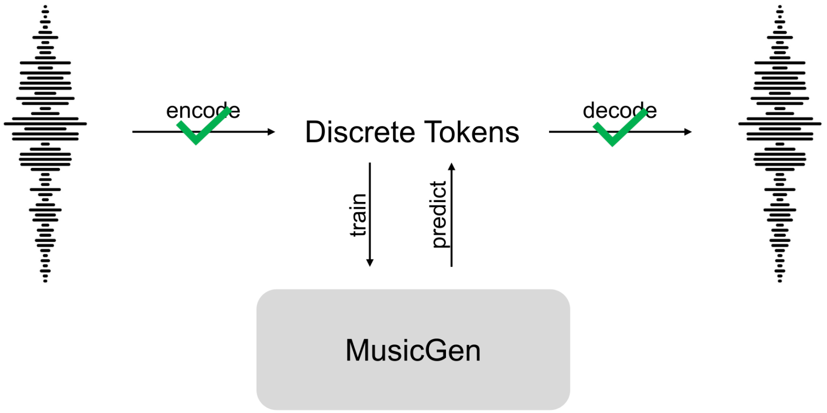


Figure 4.11: Condition on Audio Waveforms [11]

a way to parallelize the prediction of quantized values across time steps and codebooks. The ”delay” pattern from prior work [169] is used, resulting in 1500 autoregressive steps for 30 seconds of audio.

The audio tokenization model used in MusicGen is EnCodec [5], a non-causal five layers model for 32 kHz monophonic audio with a stride of 640, resulting in a frame rate of 50 Hz, and an initial hidden size of 64, doubling at each of the model's five layers. The embeddings are quantized with a RVQ with four quantizers, each with a codebook size of 2048. The model is trained on one-second audio segments cropped at random in the audio sequence.

The input modalities can be audio waveforms, chromagrams, or text. The MusicGen model uses different encoders to map the input modalities into a sequence of discrete tokens, which are then used as input to the model. The encoders used for each modality are described in the following sections.

4.3.2 Conditioning the Model

4.3.2.1 Audio Waveform Conditioning

When conditioning MusicGen on audio waveforms, the authors use an encoder called EnCodec to map the raw audio waveform into a sequence of discrete tokens (Figure 4.11). These tokens are then used as input to MusicGen, which generates new music samples based on the learned patterns in the input sequence. The EnCodec encoder uses a convolutional autoencoder architecture with a latent space quantized using Residual Vector

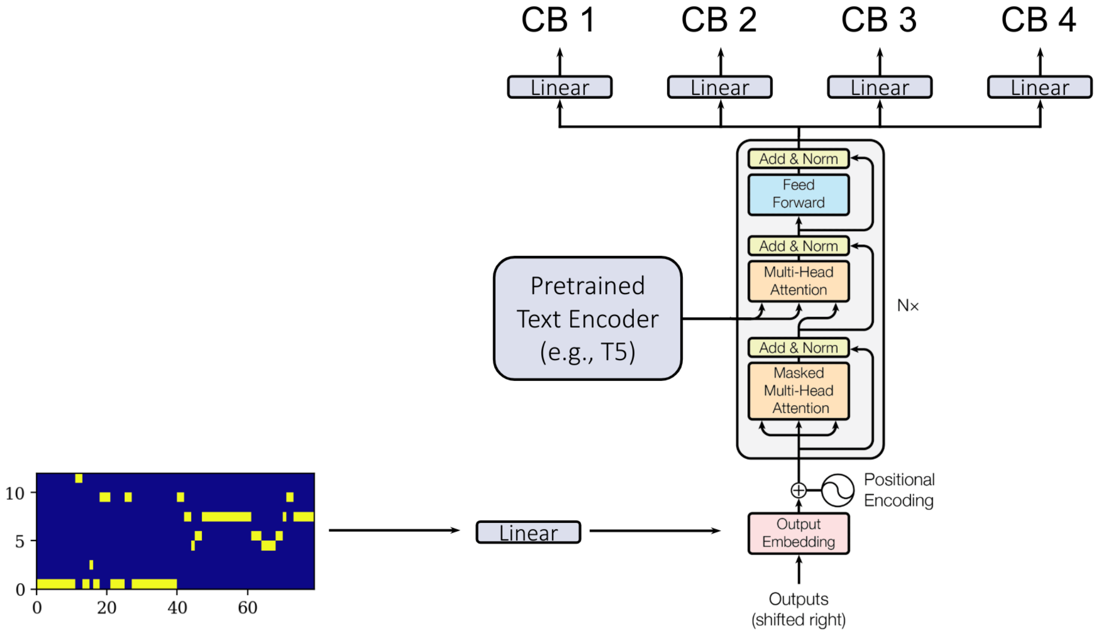


FIGURE 4.12: **C**ondition on Chromagrams [11]

Quantization (RVQ). The encoder takes in raw audio waveforms and outputs a sequence of discrete tokens, which are then used as input to MusicGen.

4.3.2.2 Chromagram Conditioning

When conditioning MusicGen on chromagrams, the authors use a quantizer to map the chromagram into a sequence of discrete tokens (Figure 4.12). These tokens are then used as input to MusicGen, which generates new music samples based on the learned patterns in the input sequence. The quantizer maps the chromagram into a sequence of discrete tokens by quantizing each chroma bin into one of a fixed number of levels. The resulting sequence of discrete tokens is then used as input to MusicGen.

The chromagrams are computed with a window size of 2^{14} and a hop size of 2^{12}, and quantized by taking the argmax at each time step. This unsupervised approach avoids the need for supervised proprietary data. During training, the condition is dropped with a probability of 0.2, and during inference, classifier-free guidance [69] is used with a guidance scale of 3.0.

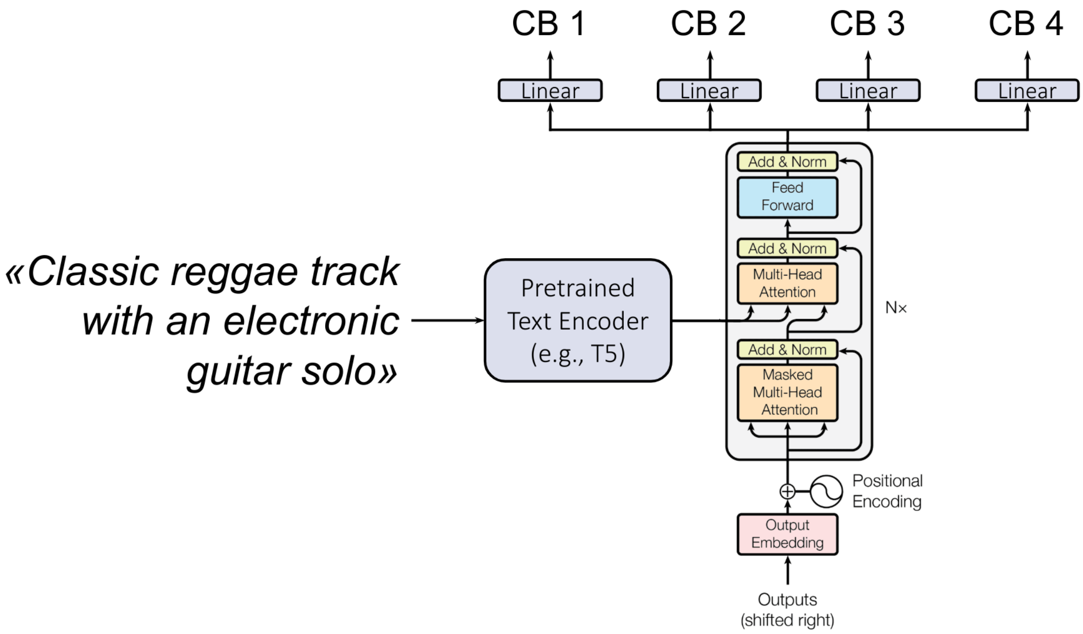


FIGURE 4.13: Condition on Text [11]

4.3.2.3 Text Conditioning

When conditioning MusicGen on text, the authors use a pre-trained language model called T5 to encode the text into a sequence of discrete tokens (Figure 4.13). These tokens are then used as input to MusicGen, which generates new music samples based on the learned patterns in the input sequence. The T5 encoder uses a transformer architecture to encode the input text into a sequence of discrete tokens. The encoder takes in raw text and outputs a sequence of discrete tokens, which are then used as input to MusicGen.

For text conditioning, a pretrained text encoder is used to obtain a conditioning tensor $C \in \mathbb{R}^{T_C \times D}$, where D is the inner dimension of the autoregressive model. The authors experiment with different text encoders: T5 [170], FLAN-T5 [171], and CLAP [172].

4.3.3 Training and Evaluation

The model was trained on 20K hours of licensed music data, including an internal dataset of 10K high-quality tracks, and the Shutterstock and Pond5 music collections with 25K and 365K instrument-only tracks, respectively. The audio is downmixed to mono unless otherwise specified. For evaluation, the authors used the MusicCaps benchmark [59], consisting of 5.5K ten-second samples with textual descriptions, and a 1K subset balanced across genres for qualitative evaluation. Objective metrics used were the Fréchet Audio Distance (FAD), Kullback-Leibler divergence (KL) over AudioSet labels, and the CLAP score [172] for audio-text alignment.

4.4 Reconstructing Music from Cortical Voxel Time-series

The core objective of this work is to reconstruct music stimuli from cortical voxel time-series, which represent the brain activity of subjects while listening to music. This section describes the methodological framework employed to achieve this goal, building upon the foundation laid in the previous sections regarding the extraction of cortical voxel time-series from raw fMRI scans and the utilization of the MusicGen model for conditional music generation.

The reconstruction process involves mapping the cortical voxel time-series, which capture the temporal patterns of neural activity in different brain regions, to the prior embedding space of the MusicGen model. This mapping enables the generation of reconstructed music stimuli by conditioning the MusicGen model on the predicted prior embeddings. We commence by introducing the Map Method [164], a general theoretical approach for solving Brain-Conditional Multimodal Synthesis tasks, including the reconstruction of music from brain signals. The Map Method involves learning a mapping from the cortical voxel time-series to the prior space of a pretrained generative model, leveraging the powerful generative capabilities of these models while focusing the learning task on the more tractable problem of mapping brain signals to priors.

However, a key challenge arises due to the potential mismatch between the temporal dimensions of the cortical voxel time-series and the prior embedding space. To address this, we explore various temporal alignment techniques, including sliding window averaging, skipped timesteps, and total averaging, to ensure compatibility between the input and output representations.

Building upon the Map Method and temporal alignment strategies, we present a mapping ensemble model that leverages the localized nature of brain function organization. This approach independently models the relationship between the voxel time-series of each anatomically defined region of interest (ROI) and the target prior embedding space. The predictions from the top-performing ROIs, as determined by their correlation scores with the ground truth embeddings, are combined through an ensemble averaging process to obtain a more accurate and robust overall mapping.

The following subsections delve into the technical details of the Map Method, the temporal alignment problem and solutions, and the mapping ensemble model, providing a comprehensive understanding of the methodological framework for reconstructing music from cortical voxel time-series.

4.4.1 The Map Method

The Map Method [164] is a general theoretical approach for solving Brain-Conditional Multimodal Synthesis tasks, which aim to decode brain signals back to perceptual experiences across different modalities (Figure 4.14). This method involves mapping brain signals to the prior space (semantic or detail priors) of a pretrained AIGC (AI Generated Content) decoder model. The Map Method has gained popularity due to its advantages of easy training, flexible implementation, and high fidelity of the generated content. Numerous works have adopted the Map Method for various Brain-Conditional Multimodal Synthesis tasks, including:

- Image-Brain-Image (IBI): [79, 173, 174, 80, 175, 176, 177, 178, 179, 87, 180, 181, 182, 183, 81, 184, 185, 78, 186, 187, 188, 189, 190, 191, 192]
- Video-Brain-Video (VBV): [193]
- Speech-Brain-Speech (SBS): [88]
- Music-Brain-Music (MBM): [95]
- Image-Brain-Text (IBT): [174, 194, 195, 196, 197]
- Video-Brain-Text (VBT) and Speech-Brain-Text (SBT): [92]

It involves mapping the cortical voxel time-series, which represent the brain activity, to the prior space of a pretrained generative model for music synthesis. Let $\mathcal{G}$ be a pretrained conditional generative model for music synthesis, capable of generating music g conditioned on input priors p from a prior space $\mathcal{P}$:

$$g = \mathcal{G}(p), \quad p \in \mathcal{P} \tag{4.4}$$

Let $\mathbf{X} = \{\boldsymbol{x_v} | \boldsymbol{v} \in \mathcal{V}\}$ represent the fMRI data, where $\boldsymbol{x_v} \in \mathbb{R}^T$ is the time-series of voxel intensities at spatial coordinate $\boldsymbol{v}$ over T time steps. We assume that the brain can be divided into N functional regions $\mathcal{R} = \mathcal{R}ii = 1^N$, where each region $\mathcal{R}_i$ is associated with a set of voxel coordinates $\mathcal{V}_i \subseteq \mathcal{V}$. The voxel time-series within each region can be represented as $\mathbf{A}i = \{\boldsymbol{x_v} | \boldsymbol{v} \in \mathcal{V}_i\}$. The Map Method aims to learn a mapping $\Phi_i : \mathbf{A}_i \rightarrow \mathcal{P}$ for each brain region $\mathcal{R}_i$, which maps the voxel time-series $\mathbf{A}i$ to the prior space $\mathcal{P}$ of the generative model $\mathcal{G}$. This mapping is parameterized by a function $f\theta_i$ with parameters θ_i, and is learned from a training set $\mathcal{D}_i = \mathbf{A}i^j, p^j j = 1^{M_i}$, where $\mathbf{A}_i^j$ are the voxel time-series in region $\mathcal{R}_i$ when listening to music sample j, and p^j is the corresponding ground truth prior for that music sample. The objective is to minimize a loss function $\mathcal{L}_i$ between the predicted priors $\Phi_i(\mathbf{A}_i^j)$ and the ground truth priors p^j:

$$\theta_i^* = \arg\min_{\theta_i} \sum_{j=1}^{M_i} \mathcal{L}_i(\Phi_i(\mathbf{A}_i^j), p^j) \tag{4.5}$$

The learned mapping $\Phi_i = f_{\theta_i^*}$ can then be applied to unseen voxel time-series $\mathbf{A}_i'$ in region $\mathcal{R}_i$ to predict the corresponding prior $\Phi_i(\mathbf{A}_i')$, enabling the generation of reconstructed music g' via the generative model $\mathcal{G}$:

$$g' = \mathcal{G}(\Phi_i(\mathbf{A}_i')) \tag{4.6}$$

In practice, an ensemble of mappings $\Phi_{ii=1}^{N}$ is learned for different brain regions, and their predictions are combined (e.g., averaged) to obtain the final reconstructed music. The Prior space $\mathcal{P}$ is of MusicGen which can be EnCodec, Chromagram or T5 tokens.

The core advantage of the Map Method is that it simplifies the complex task of directly reconstructing modalities from brain signals by introducing an intermediate mapping step. Instead of learning a direct mapping from brain signals to the target modality, the Map Method learns a mapping from brain signals to the prior space of a pretrained generative model. This allows leveraging powerful generative models that have been trained on large datasets, while focusing the learning task on the mapping from brain signals to priors, which can be easier to optimize.

However, the Map Method also has a potential limitation: biases introduced in the mapping space can propagate and amplify in the generation space, leading to semantic ambiguity in the reconstructed content. This issue, known as bias superposition, arises because the Map Method only constructs the brain-prior connection and relies on the generative model to decode the priors into the target modality.

Despite this limitation, the Map Method remains a popular choice for Brain-Conditional Multimodal Synthesis due to its advantages of easy training, flexible implementation, and high fidelity of the generated content when combined with powerful generative models.

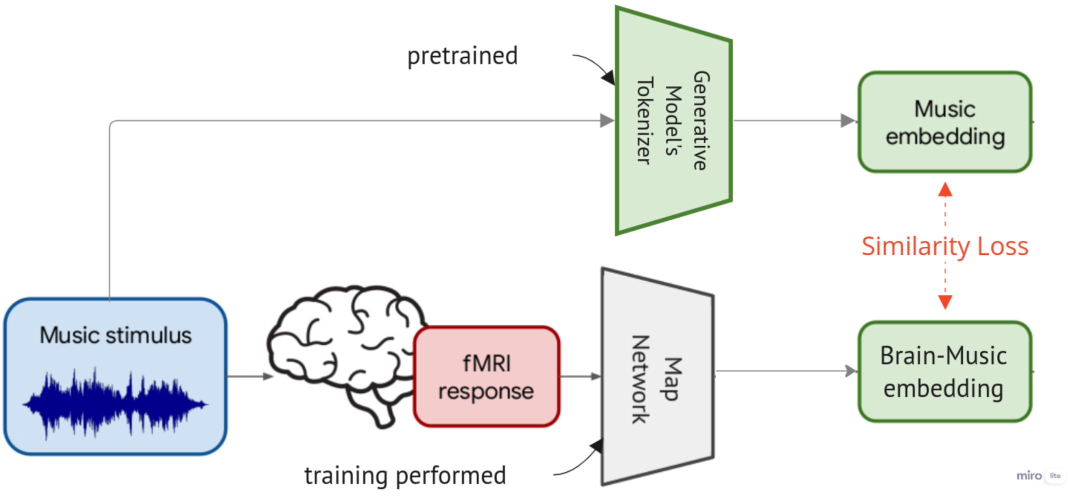


FIGURE 4.14: An overview of Map Method as applied to Music Generation. The Generative model's encoder/tokenizers extract prior music embeddings. A mapping function/network is learned to minimize the similarity loss between the mapped brain-music embedding and the prior music embeddings. Thus the mapped embeddings can condition music generation models to reconstruct the listened music.

4.4.2 Temporal Alignment Problem

The Map Method involves learning a mapping $\Phi : B \to T$ between the fMRI response tensor $B \in \mathbb{R}^{s \times v}$ and the prior embedding space $T \in \mathbb{R}^{r \times d}$, where s is the number of fMRI scans, v is the number of voxel intensities, r is the number of embeddings computed by the encoder E_k, and d is the embedding dimension. However, a key challenge arises due to the potential mismatch between the temporal dimensions s and r. The number of fMRI scans s is fixed and determined by the experimental setup, whereas the number of embeddings r varies and depends on the specific encoder E_k. In many cases, r is larger than s, as the encoder may compute embeddings at a higher temporal resolution than the fMRI scans. To enable the learning of the mapping model Φ, a temporal alignment between s and r is necessary. This can be achieved by either downsampling r to match s or downsampling both s and r to a common smaller dimension.

In our work, we experimented with three techniques for temporal alignment (Figure 4.15):

Sliding Window Averaging: In this approach, the entries in T along the r dimension are averaged using a sliding window operation to match the time ranges of a single fMRI scan in B. Specifically, let w be the window size, and $T_i \in \mathbb{R}^{w \times d}$ be the i-th window of w consecutive embeddings in T. The sliding window averaging operation computes a new embedding $\bar{T}_i \in \mathbb{R}^d$ as follows:

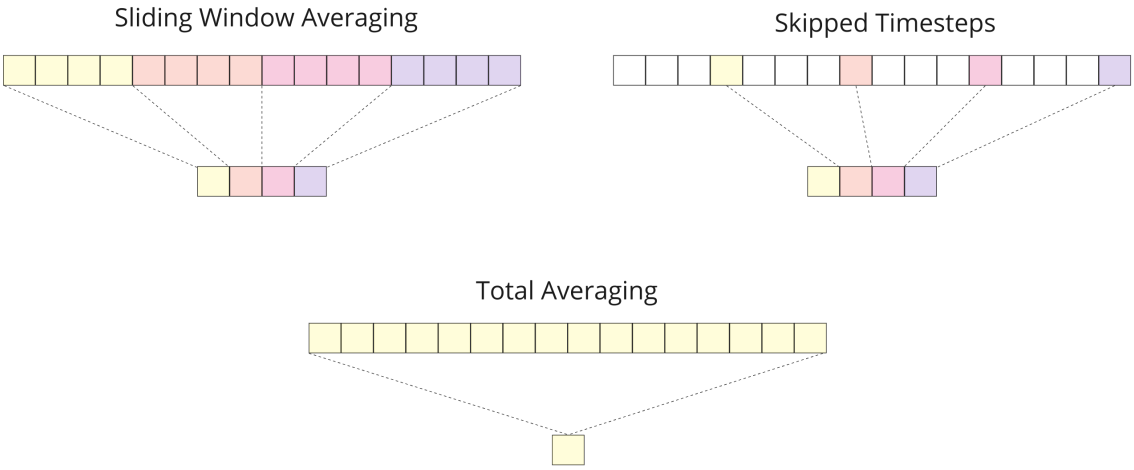


FIGURE 4.15: Temporal Alignment Techniques: Sliding window averaging (top-left), skipped timesteps (top-right) and total averaging (bottom)

$$\bar{T}_i = \frac{1}{w} \sum_{j=1}^{w} T_{i,j} \tag{4.7}$$

This process is repeated in a sliding window fashion across the entire embedding vector T, resulting in a downsampled embedding vector $\bar{T} \in \mathbb{R}^{\lfloor r/w \rfloor \times d}$ that aligns with the number of fMRI scans s.

Skipped Timesteps: Instead of averaging the entries in T, this approach selects equidistant entries in T to align with the number of fMRI scans s. Let $k = \lfloor r/s \rfloor$ be the stride size, and $T_i \in \mathbb{R}^d$ be the i-th entry in T. The skipped timesteps operation constructs a new embedding vector $\hat{T} \in \mathbb{R}^{s \times d}$ as follows:

$$\hat{T}_i = T_{i \times k} \tag{4.8}$$

Total Averaging: Rather than aligning r and s, this approach averages both the embedding vector T and the fMRI response tensor B to obtain a single vector representation. The total averaging operation computes a scalar embedding $\tilde{T} \in \mathbb{R}^d$ and a scalar fMRI response $\tilde{B} \in \mathbb{R}^v$ as follows:

$$\tilde{T} = \frac{1}{r} \sum_{i=1}^{r} T_i, \quad \tilde{B} = \frac{1}{s} \sum_{i=1}^{s} B_i \tag{4.9}$$

The mapping Φ is then learned to map the fMRI response $\tilde{B}$ to the aligned embedding $\tilde{T}$. These temporal alignment techniques enable the Map Method to handle the potential mismatch between the temporal dimensions of the fMRI response tensor and the prior embedding space, facilitating the learning of the mapping model Φ. The choice of technique may depend on factors such as the desired temporal resolution, computational efficiency, and the specific characteristics of the data.

4.4.3 Mapping Ensemble Model

The Map Method aims to learn a mapping $\Phi : B \rightarrow T$ between the fMRI response tensor $B \in \mathbb{R}^{s \times v}$ and the prior embedding space $T \in \mathbb{R}^{r \times d}$, where s is the number of fMRI scans, v is the number of voxel intensities, and d is the embedding dimension. As discussed earlier, the fMRI data $\mathbf{X}$ can be divided into N brain regions $\mathbf{A}_{i}{}_{i=0}^{N-1}$ based on the principle of localization in brain function organization:

$$\mathbf{X} = \bigcup_{i=0}^{N-1} \mathbf{A}_i, \tag{4.10}$$

where $\mathbf{A}_i = \{\boldsymbol{x_v} | \boldsymbol{v} \in \mathbf{V}_i\}$ represents the set of voxel time series in the i-th brain region, with $\mathbf{V}_i$ denoting the set of three-dimensional coordinates for that region. Leveraging this brain region segmentation, we independently model the relationship between the voxel time series $\mathbf{B} + \text{roi} \in \mathbb{R}^{s \times v\text{roi}}$ of a specific region of interest (ROI) and the aligned target prior embedding $T \in \mathbb{R}^{s \times d}$, where v_{roi} is the number of voxels in the ROI. The mapping for the ROI is parameterized by a weight matrix $\mathbf{W}_{\text{roi}} \in \mathbb{R}^{v_{\text{roi}} \times d}$, and the predicted embedding $\hat{T}_{\text{roi}} \in \mathbb{R}^{s \times d}$ is computed as:

$$\hat{T}_{\text{roi}} = \mathbf{B}_{\text{roi}} \mathbf{W}_{\text{roi}} \tag{4.11}$$

We use an L2-regularized linear regression to estimate the weight matrix $\mathbf{W}_{\text{roi}}$ on the training dataset $\mathcal{D} = \{(\mathbf{B}_{\text{roi}}^{(i)}, T^{(i)})\}_{i=1}^{M}$, where $\mathbf{B}_{\text{roi}}^{(i)} \in \mathbb{R}^{s \times v_{\text{roi}}}$ is the voxel time series for the ROI when listening to the i-th music sample, and $T^{(i)} \in \mathbb{R}^{s \times d}$ is the corresponding ground truth prior embedding. The objective is to minimize the following regularized loss function:

$$\mathcal{L}(\mathbf{W}_{\text{roi}}) = \frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{B}_{\text{roi}}^{(i)} \mathbf{W}_{\text{roi}} - T^{(i)} \right\|_F^2 + \lambda \left\| \mathbf{W}_{\text{roi}} \right\|_F^2 \tag{4.12}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\lambda > 0$ is the regularization parameter controlling the trade-off between data fitting and model complexity.

The training process is independently performed for each anatomically defined ROI in the Destrieux Atlas [7], which segments the brain into 148 ROIs. After training, we select a subset of top-performing ROIs based on their correlation scores between the predicted embeddings $\hat{T}_{\text{roi}}$ and the ground truth embeddings T, as determined via cross-validation on the training data. To combine the predictions from the selected top ROIs, we create an ensemble model by averaging their predicted embeddings. Specifically, let $\mathcal{C} \subseteq 0, 1, \ldots, 147$ be the set of indices of the selected top ROIs. For an unseen fMRI response tensor $B_s \in \mathbb{R}^{148 \times s \times v}$, the ensemble prediction $\hat{T}_s \in \mathbb{R}^{s \times d}$ is computed as:

$$\hat{T}_s = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \mathbf{B}_{\text{roi}_i} \mathbf{W}\text{roi}_i \tag{4.13}$$

where $\mathbf{B}_{\text{roi}_i} \in \mathbb{R}^{s \times v_{\text{roi}_i}}$ is the voxel time series for the i-th selected ROI, and $\mathbf{W}_{\text{roi}_i} \in \mathbb{R}^{v_{\text{roi}_i} \times d}$ is the corresponding learned weight matrix. The ensemble model leverages the complementary information captured by different brain regions, potentially improving the overall mapping accuracy and robustness.

The final reconstructed music stimulus $\mathcal{K}'_s$ can then be generated via the conditional music generation model $\mathcal{G}$ using the ensemble prediction $\hat{T}_s$ as input:

$$\mathcal{K}'_s = \mathcal{G}(\hat{T}_s) \tag{4.14}$$

The mapping ensemble model approach takes advantage of the localized nature of brain function organization by independently modelling the mapping from voxel time series to prior embeddings for each ROI, and subsequently combining the predictions from the top-performing ROIs to obtain a more accurate and robust ensemble mapping.

Chapter 5

Experimental Setup

5.1 Dataset

The music fMRI dataset used in this study comes from the *music genre neuroimaging dataset*[1] by [50]. The dataset contains music stimuli from 10 genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. These stimuli were sampled randomly from the GTZAN dataset [198]. In total, 54 music pieces of 30 seconds duration at 22.050kHz sampling rate were selected from each genre, resulting in 540 music pieces.

For each music piece, a 15-second clip was randomly extracted. Each clip was subjected to 2 seconds of fade-in and fade-out effects, and the overall signal intensity was normalized. The dataset is split into 480 examples for training and 60 examples for testing, with no repetition in the training set.

Data Collection During the fMRI scanning process, five participants were instructed to focus on a fixation cross at the center of the screen while listening to the music clips through MRI-compatible insert earphones (Model S14, Sensimetrics). This headphone model can attenuate scanner noise and has been widely used in previous MRI studies with auditory stimuli [199]. Scanning was performed using a 3.0T MRI scanner (TIM Trio; Siemens, Erlangen, Germany) equipped with a 32-channel head coil. For functional scanning, 68 interleaved axial slices with a thickness of 2.0mm were scanned without a gap using a T2*-weighted gradient echo multi-band echo-planar imaging (MB-EPI) sequence [200]. The scanning parameters were as follows: repetition time (TR) = 1,500ms, echo

[1] `https://openneuro.org/datasets/ds003720/versions/1.0.0`

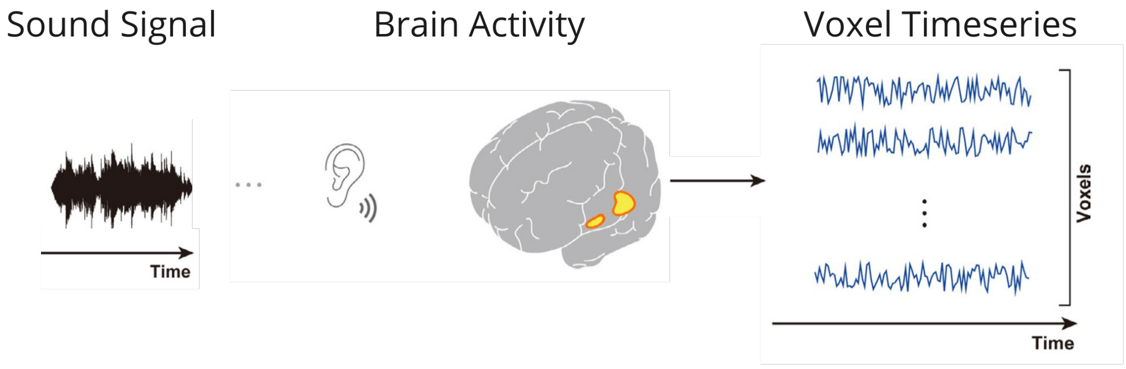


FIGURE 5.1: Overview of how Voxel Timeseries are generated in response to music stimuli

time (TE) = 30ms, flip angle (FA) = 62°, field of view (FOV) = 192 × 192 mm^2, voxel size = 2 × 2 × 2 mm^3, and multi-band factor = 4. A total of 410 volumes were obtained for each run.

The fMRI data $\mathbf{X}$ can be represented as a set of time series

$$\mathbf{X} = \{\boldsymbol{x_v} | \boldsymbol{v}^{(0)} \in [0, W), \boldsymbol{v}^{(1)} \in [0, H), \boldsymbol{v}^{(2)} \in [0, D)\}. \tag{5.1}$$

where $\boldsymbol{v} = [\boldsymbol{v}^{(0)}, \boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}]$ represents the 3D spatial coordinates, and $\boldsymbol{xv} \in \mathbb{R}^T$ represents the fMRI signal time series at the corresponding spatial location with W, H, D, T denoting the width, height, depth, and length of the time series, respectively. Each subject listened to 41 music samples in a single run of approximately 10 minutes duration. In total, there are 18 runs, with 12 runs in the training set and 6 runs in the test set. Each run consists of 410 T2-weighted EPI images (4D), with each image taken at a 1.5-second interval. Thus, for each 15-second music sample, there are 10 corresponding fMRI scans. Additionally, for each subject, a T1-weighted anatomical scan (3D) is available.

Text Captions The original dataset by [50] has been augmented with English text captions[2], which describe the musical pieces in terms of genre, instrumentation, rhythm, and mood. These captions average approximately 46 words or 280 characters in length and often comprise fragmented or semi-complete sentences, with an average of about 4.5 sentences per caption. The writing style is subjective, reflecting not only the technical components of the music but also the emotional responses or atmospheres they might evoke in listeners. The captions were originally written in Japanese and translated using DeepL.

[2] `https://www.kaggle.com/datasets/nishimotolab/music-caption-brain2music`

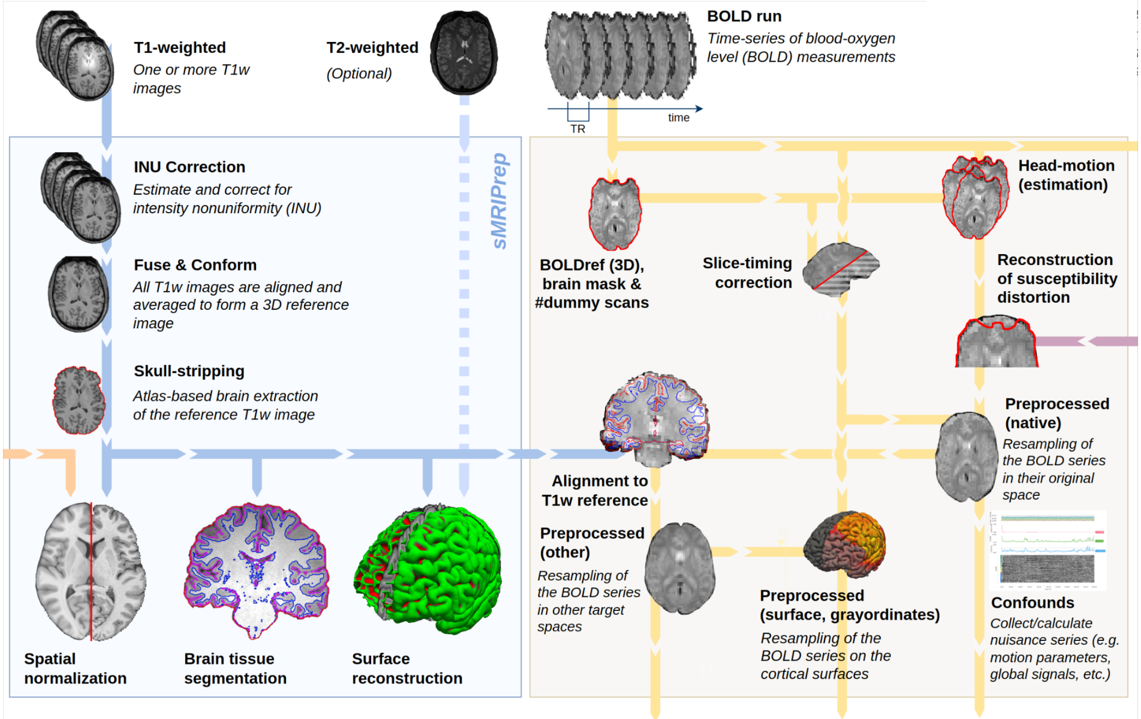

T1-weighted
One or more T1w images
T2-weighted
(Optional)
BOLD run
Time-series of blood-oxygen level (BOLD) measurements
TR
time
INU Correction
Estimate and correct for intensity nonuniformity (INU)
Fuse & Conform
All T1w images are aligned and averaged to form a 3D reference image
Skull-stripping
Atlas-based brain extraction of the reference T1w image
sMRIPrep
BOLDref (3D), brain mask & #dummy scans
Slice-timing correction
Head-motion (estimation)
Reconstruction of susceptibility distortion
Preprocessed (native)
Resampling of the BOLD series in their original space
Alignment to T1w reference
Preprocessed (other)
Resampling of the BOLD series in other target spaces
Preprocessed (surface, grayordinates)
Resampling of the BOLD series on the cortical surfaces
Confounds
Collect/calculate nuisance series (e.g. motion parameters, global signals, etc.)
Spatial normalization
Brain tissue segmentation
Surface reconstruction

FIGURE 5.2: fMRIPrep [12] PreprocessingPipeline

5.2 Data Preprocessing; fMRIPrep

The raw fMRI data were preprocessed using the fMRIPrep software package [12], which adapts its pipeline based on the available data and metadata (Figure 5.2). The preprocessing workflow for structural MRI begins by constructing an average image from all available T1-weighted (T1w) images, conforming them to RAS orientation and a common voxel size. Brain extraction, tissue segmentation, and spatial normalization to standard spaces are then performed on this averaged T1w reference.

A crucial aspect of the structural MRI preprocessing is surface reconstruction, which is carried out using FreeSurfer [201]. This process involves initializing the subject with T1w and T2w (if available) structural images, performing basic reconstruction with skull-stripping skipped, importing the previously calculated brain mask, and resuming reconstruction while utilizing the T2w image to assist in locating the pial surface. The reconstructed white and pial surfaces are included in the final report.

For BOLD (T2*-weighted echo-planar imaging, EPI) data preprocessing, the workflow is divided into fit and transform stages. Initially, a reference image for each BOLD series is estimated, either by averaging non-steady-state volumes or taking the median of motion-corrected volumes. Head-motion estimation is then performed using FSL's mcflirt [202], and slice-timing correction is applied if the necessary metadata is available.

A critical step in the BOLD preprocessing pipeline is the EPI to T1w registration, which aligns the reference EPI image of each run to the reconstructed subject's gray/white matter boundary obtained from FreeSurfer's surfaces. This registration was performed using FreeSurfer's bbregister routine which uses the Boundary Based Registration (BBR) cost function.

Structural and Functional Parcellations To obtain functional parcellations and extract voxel time-series from specific regions of interest (ROIs), the structural parcellations derived from FreeSurfer's Destrieux Atlas [7] are transformed into the coordinate space of the functional mean reference image for each BOLD run. This transformation is facilitated by the previously performed EPI to T1w registration, which aligns the EPI data to the reconstructed subject's structural space. By mapping the atlas parcellations onto the functional EPI images, we can extract the voxel time-series corresponding to each of the 148 ROIs defined in the Destrieux Atlas. These voxel time-series serve as the basis for subsequent analyses, allowing us to investigate the relationship between brain activity and the perceived auditory stimuli.

The table 5.1 provides the number of extracted voxels in each ROI defined in the Destrieux Atlas, illustrating the extent of the parcellations obtained from the structural MRI data and subsequently mapped onto the functional EPI images. It can be noticed that ROIs can vary drastically in their size and also across the different hemispheres. Every individual has a unique anatomical structure of the brain.

5.3 Implementation Details

The preprocessing of fMRI data was performed using fMRIPrep [12] and FreeSurfer [201]. Due to the computational constraints of running fMRIPrep, which takes approximately 24 hours to process a single subject on a 10th generation Intel Core i7 processor with 32 GB RAM, our analysis and experiments focused on data from subject 1.

For music generation, we utilized MusicGen [11], a simple and controllable model provided by Meta's Audiocraft. MusicGen is a single-stage auto-regressive Transformer model

Table 5.1: Extracted Number of Voxels in each region of interest in Destriuex Atlas

Index	Left Hemisphere ROI	Voxels	Right Hemisphere ROI	Voxels
1	lh_G_temporal_middle	907	rh_G_temporal_middle	1204
2	lh_G_temp_sup-Lateral	756	rh_G_temp_sup-Lateral	671
3	lh_G_temporal_inf	1129	rh_G_temporal_inf	835
4	lh_S_temporal_sup	1204	rh_S_temporal_sup	1184
5	lh_G_pariet_inf-Supramar	775	rh_G_pariet_inf-Supramar	813
6	lh_G_temp_sup-Plan_tempo	266	rh_G_temp_sup-Plan_tempo	217
7	lh_S_temporal_inf	286	rh_S_temporal_inf	250
8	lh_G_and_S_subcentral	408	rh_G_and_S_subcentral	354
9	lh_G_pariet_inf-Angular	730	rh_G_pariet_inf-Angular	733
10	lh_S_postcentral	564	rh_S_postcentral	355
11	lh_G_postcentral	427	rh_G_postcentral	391
12	lh_S_interm_prim-Jensen	21	rh_S_interm_prim-Jensen	92
13	lh_G_occipital_middle	736	rh_G_occipital_middle	540
14	lh_G_precentral	817	rh_G_precentral	715
15	lh_S_oc_middle_and_Lunatus	137	rh_S_oc_middle_and_Lunatus	196
16	lh_S_central	488	rh_S_central	422
17	lh_G_front_inf-Opercular	461	rh_G_front_inf-Opercular	408
18	lh_G_temp_sup-Plan_polar	229	rh_G_temp_sup-Plan_polar	283
19	lh_G_and_S_occipital_inf	359	rh_G_and_S_occipital_inf	375
20	lh_S_oc-temp_lat	117	rh_S_oc-temp_lat	209
21	lh_Lat_Fis-post	219	rh_Lat_Fis-post	228
22	lh_S_collat_transv_ant	253	rh_S_collat_transv_ant	194
23	lh_S_temporal_transverse	85	rh_S_temporal_transverse	59
24	lh_S_precentral-inf-part	322	rh_S_precentral-inf-part	305
25	lh_S_occipital_ant	180	rh_S_occipital_ant	151
26	lh_G_temp_sup-G_T_transv	129	rh_G_temp_sup-G_T_transv	90
27	lh_Pole_temporal	788	rh_Pole_temporal	756
28	lh_G_front_middle	1133	rh_G_front_middle	1093
29	lh_Lat_Fis-ant-Vertical	43	rh_Lat_Fis-ant-Vertical	54
30	lh_G_front_inf-Triangul	255	rh_G_front_inf-Triangul	339
31	lh_S_intrapariet_and_P_trans	459	rh_S_intrapariet_and_P_trans	579
32	lh_G_oc-temp_lat-fusifor	475	rh_G_oc-temp_lat-fusifor	521
33	lh_S_circular_insula_inf	341	rh_S_circular_insula_inf	236
34	lh_S_oc_sup_and_transversal	283	rh_S_oc_sup_and_transversal	181
35	lh_S_circular_insula_sup	321	rh_S_circular_insula_sup	239
36	lh_G_parietal_sup	664	rh_G_parietal_sup	506
37	lh_S_front_inf	273	rh_S_front_inf	279
38	lh_G_front_inf-Orbital	110	rh_G_front_inf-Orbital	123
39	lh_S_precentral-sup-part	241	rh_S_precentral-sup-part	223
40	lh_S_orbital_lateral	29	rh_S_orbital_lateral	92
41	lh_G_orbital	703	rh_G_orbital	803
42	lh_G_Ins_lg_and_S_cent_ins	190	rh_G_Ins_lg_and_S_cent_ins	191
43	lh_G_insular_short	329	rh_G_insular_short	271
44	lh_S_collat_transv_post	75	rh_S_collat_transv_post	60
45	lh_S_oc-temp_med_and_Lingual	346	rh_S_oc-temp_med_and_Lingual	312
46	lh_Lat_Fis-ant-Horizont	37	rh_Lat_Fis-ant-Horizont	49
47	lh_G_oc-temp_med-Parahip	424	rh_G_oc-temp_med-Parahip	417
48	lh_S_front_sup	714	rh_S_front_sup	442
49	lh_G_and_S_frontomargin	296	rh_G_and_S_frontomargin	184
50	lh_S_calcarine	394	rh_S_calcarine	382
51	lh_G_front_sup	2506	rh_G_front_sup	1868
52	lh_S_front_middle	248	rh_S_front_middle	482
53	lh_Pole_occipital	310	rh_Pole_occipital	560
54	lh_G_occipital_sup	333	rh_G_occipital_sup	382
55	lh_S_orbital-H_Shaped	277	rh_S_orbital-H_Shaped	266
56	lh_G_oc-temp_med-Lingual	521	rh_G_oc-temp_med-Lingual	526
57	lh_S_circular_insula_ant	97	rh_S_circular_insula_ant	124
58	lh_S_parieto_occipital	388	rh_S_parieto_occipital	391
59	lh_G_and_S_transv_frontopol	218	rh_G_and_S_transv_frontopol	344
60	lh_G_cuneus	364	rh_G_cuneus	372
61	lh_G_and_S_paracentral	365	rh_G_and_S_paracentral	307
62	lh_G_subcallosal	98	rh_G_subcallosal	88
63	lh_S_cingul-Marginalis	179	rh_S_cingul-Marginalis	267
64	lh_G_precuneus	700	rh_G_precuneus	650
65	lh_G_cingul-Post-ventral	94	rh_G_cingul-Post-ventral	97
66	lh_S_orbital_med-olfact	100	rh_S_orbital_med-olfact	131
67	lh_S_subparietal	154	rh_S_subparietal	222
68	lh_G_and_S_cingul-Mid-Post	302	rh_G_and_S_cingul-Mid-Post	300
69	lh_S_pericallosal	166	rh_S_pericallosal	184
70	lh_G_and_S_cingul-Mid-Ant	304	rh_G_and_S_cingul-Mid-Ant	335
71	lh_G_and_S_cingul-Ant	464	rh_G_and_S_cingul-Ant	763
72	lh_G_rectus	265	rh_G_rectus	295
73	lh_S_suborbital	141	rh_S_suborbital	60
74	lh_G_cingul-Post-dorsal	197	rh_G_cingul-Post-dorsal	201

trained over a 32kHz EnCodec tokenizer with 4 codebooks sampled at 50 Hz. By introducing a small delay between the codebooks, MusicGen can predict them in parallel, resulting in only 50 auto-regressive steps per second of audio. The model architecture consists of an EnCodec model for audio tokenization and an auto-regressive language model based on the Transformer architecture for music modeling. MusicGen offers different model sizes (300M, 1.5B, and 3.3B parameters) and two variants: one for text-to-music generation and another for melody-guided music generation. We specifically used the 1.5 billion parameter ”melody” checkpoint, as it allows conditioning on chromagrams, unlike the other checkpoints.

Three modality encoders were employed in our experiments: EnCodec, Chromagram Tokenizer, and T5 [4]. For a 15-second music sample, EnCodec produces a discrete output of size [4, 517], where 4 represents the number of codebooks and 517 is the downsampled sampling rate. The Chromagram Tokenizer yields a discrete output of size [235, 1536], with 235 being the number of computed chromagrams and 1536 the embedding size of each chromagram. Additionally, we have text captions for the 15-second music clips, which vary in length. The output of T5 for a caption is of size n × 1536, where n is the number of word tokens in the caption, and T5 uses its own tokenizer. Table 5.2 summarizes the output dimensions of the different prior encoders.

Table 5.2: Output dimensions of different prior encoders.

Encoder	**Output Dimensions**
EnCodec	[4, 517]
Chromagram Tokenizer	[235, 1536]
T5 (Text Encoder)	[n,1536]

To address the temporal alignment problem present in all three encoders, we experimented with sliding window averaging, skipping timesteps, and total averaging techniques.

For learning the mapping between cortical voxel time series and prior embeddings computed by the encoders, we employed L2 Regularized Linear Regression using the Himalaya library [203]. A separate model was fit independently for each region of interest (ROI). The top 6 ROIs were then selected based on their correlation scores after 5-fold cross-validation on training data, forming an ensemble model whose outputs are averaged to produce a single prior embedding for conditioning the MusicGen model, depending on the type of prior.

All experiments were conducted using Google Colab with a T4 GPU, leveraging open-source datasets and tools. We express our gratitude to the open-source community for making this research possible.

Hyperparameter Tuning We use a ridge regression to estimate our model parameters. Citing the *himalaya* documentation[3]: Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a feature matrix with n samples and p features, $\boldsymbol{y} \in \mathbb{R}^n$ a target vector, and $\alpha > 0$ a fixed regularization hyperparameter. Ridge regression [204] defines the weight vector $\boldsymbol{b}^* \in \mathbb{R}^p$ as:

$$\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}} \|\boldsymbol{X}\boldsymbol{b} - \boldsymbol{y}\|_2^2 + \alpha \|\boldsymbol{b}\|_2^2 \,. \tag{5.2}$$

The equation has a closed-form solution $\boldsymbol{b}^* = \boldsymbol{M}\boldsymbol{y}$, where $\boldsymbol{M} = (\boldsymbol{X}^\top \boldsymbol{X} + \alpha \boldsymbol{I}_p)^{-1} \boldsymbol{X}^\top \in \mathbb{R}^{p \times n}$.

To determine α we run 5-fold cross-validation on the training data. Note that there is one α parameter per regression targets i.e. it can be 4 in the case of EnCodec or 1536 in the case of Chromagram and T5. We inspect the performance on training and evaluation data around the chosen α vector "α (opt)" in Figure 5.3.

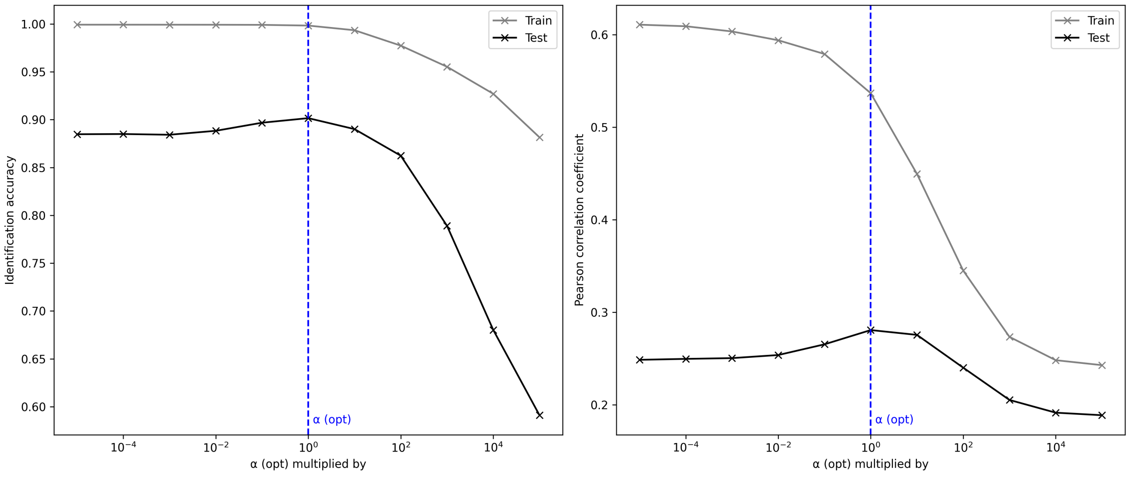


Figure 5.3: Performance of the regressor when trained with α values in the neighborhood of the α that was determined to be optimal on the *training split* via cross-validation. The model starts to overfit with lower values of α (to the left) and underfits in the opposite direction.

[3] gallantlab.org/himalaya/models.html#ridge

5.4 Evaluation Metrics

5.4.1 Prior Embedding Evaluation

To assess the performance of the linear regression model in predicting prior embeddings, we employ several evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2), and Identification Accuracy.

Mean Absolute Error (MAE). MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. Given a vector of predictions $\hat{y}$ and a vector of true values y, each containing n samples, the MAE is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{5.3}$$

MAE values range from 0 to ∞, with 0 indicating a perfect fit. Lower values are better.

Mean Squared Error (MSE). MSE measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value. Given a vector of predictions $\hat{y}$ and a vector of true values y, each containing n samples, the MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{5.4}$$

MSE values range from 0 to ∞, with 0 indicating a perfect fit. Lower values are better.

R-squared (R^2). R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Given a vector of predictions $\hat{y}$ and a vector of true values y, each containing n samples, and $\bar{y}$ being the mean of the true values, the R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}i - y_i)^2}{\sum i = 1^n (y_i - \bar{y})^2} \tag{5.5}$$

R^2 values range from 0 to 1, with 1 indicating a perfect fit. Higher values are better.

Identification Accuracy. Following the decoding literature [44, 49], we compute an *identification accuracy* of the predicted d-dimensional embeddings with respect to their target embeddings. Assume there is a matrix of predicted embeddings $\boldsymbol{P} \in \mathbb{R}^{n\times d}$ and a matrix (of equal size) containing target embeddings $\boldsymbol{T}$. Let $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ be computed from $\boldsymbol{P}$ and $\boldsymbol{T}$, specifically $\boldsymbol{C}_{i,j}$ is Pearson correlation coefficient between i-th row of $\boldsymbol{P}$ and j-th row of $\boldsymbol{T}$. The identification accuracy for the i-th prediction is defined as:

$$\text{id acc}_i = \frac{1}{n-1}\sum_{j=1}^{n} 1\left[C_{i,i} > C_{i,j}\right] \tag{5.6}$$

The identification accuracy for all examples is simply the average:

$$\text{id acc} = \frac{1}{n}\sum_{i=1}^{n} \text{id acc}_i \tag{5.7}$$

The identification accuracy, ranging from 0 to 1 with 0.5 indicating performance equivalent to random chance, provides a quantified measure of how well an embedding was predicted in relation to other embeddings in the dataset. Higher values are better.

5.4.2 Music Generation Evaluation

To evaluate the quality of the generated music samples, we compare them with the ground-truth samples using three measures: Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD), following the evaluation methodology of MusicGen [11].

Fréchet Audio Distance (FAD). FAD [205] is a measure of similarity between two distributions of audio features. It is an adaptation of the Fréchet Inception Distance (FID) [206], which is commonly used to evaluate the quality of generated images. FAD uses the VGGish [207] model, a variant of the VGG model [208] trained on audio data, to extract features from the audio samples. The VGGish model maps the audio samples into a high-dimensional feature space. The FAD is then computed as the Fréchet distance between two multivariate Gaussians fitted to the feature distributions of the real and generated audio samples:

$$\text{FAD} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right) \tag{5.8}$$

where μ_r and μ_g are the means, and Σ_r and Σ_g are the covariance matrices of the real and generated feature distributions, respectively. Tr denotes the trace of a matrix. A low

FAD score indicates that the generated audio is similar to the real audio in terms of the VGGish features, suggesting that the generated audio is plausible. FAD values range from 0 to ∞, with 0 indicating that the generated audio is indistinguishable from the ground truth. Lower values are better.

Kullback-Leibler Divergence (KL). KL divergence [209] is a measure of how one probability distribution differs from another. In the context of evaluating generated music, we use a pre-trained audio classifier to predict the probabilities of different AudioSet [210] labels for the real and generated audio samples. AudioSet is a large-scale dataset of audio events, and the pre-trained classifier is trained to predict the presence of these events in an audio sample. The KL divergence is then computed between the predicted label distributions of the real and generated audio samples:

$$\mathrm{KL}(P \parallel Q) = \sum_i P(i) \log \left(\frac{P(i)}{Q(i)} \right) \tag{5.9}$$

where P and Q are the predicted label distributions for the real and generated audio samples, respectively, and i ranges over all the AudioSet labels. The KL divergence measures how much information is lost when the label distribution of the generated audio is used to approximate the label distribution of the real audio. A low KL divergence indicates that the generated audio shares similar high-level concepts with the real audio. KL divergence values range from 0 to ∞, with 0 indicating that the generated audio has the same label distribution as the ground truth. Lower values are better.

Mel Cepstral Distortion (MCD). MCD [211] is a measure of the difference between two sequences of mel-frequency cepstral coefficients (MFCCs). MFCCs are a compact representation of the short-term power spectrum of an audio signal, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. The mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. MFCCs are commonly used as features in speech recognition and synthesis systems, as they provide a good approximation of the human auditory system's response. The MCD is computed as the Euclidean distance between the MFCC sequences of the real and generated audio samples:

$$\mathrm{MCD} = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\sum_{d=1}^{D} (\hat{c}t, d - ct, d)^2} \tag{5.10}$$

where $\hat{c}$ and c are the MFCC sequences of the generated and real audio samples, respectively, T is the number of frames, and D is the number of MFCC dimensions. A low MCD value indicates that the generated audio is similar to the real audio in terms of the mel-cepstral features, suggesting that the generated audio has a similar timbre and spectral content as the real audio. MCD values range from 0 to ∞, with 0 indicating that the generated audio is identical to the ground truth. Lower values are better. In practice, an MCD value below 4 is considered to be of good quality.

Chapter 6

Results and Discussion

In this chapter, we present the results of our experiments on predicting prior embeddings from fMRI data and reconstructing music from these predicted embeddings. We evaluate the performance of three temporal alignment techniques (sliding window averaging, skipped timesteps, and total averaging) in combination with three modality encoders (EnCodec, Chromagram Tokenizer, and T5) for both tasks. The results are compared using various evaluation metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R^2), and Identification Accuracy for prior embedding prediction, and Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD) for music reconstruction. We also discuss the scientific insights gained from analyzing the top-performing regions of interest (ROIs) in the brain for this task.

6.1 Prior Embedding Prediction Results

Table 6.1 presents the results of prior embedding prediction using different temporal alignment techniques and modality encoders. The results show that the total averaging technique consistently outperforms the other two techniques across all modality encoders, with the T5 encoder achieving the best overall performance.

The superior performance of the total averaging technique can be attributed to its ability to capture the overall characteristics of the music sample and the corresponding brain activity, while the other two techniques may lose important information due to downsampling or skipping. The T5 encoder's better performance compared to the other encoders suggests

Table 6.1: Prior embedding prediction results using different temporal alignment techniques and modality encoders. Best results for each encoder are in bold.

Method	**MAE ↓**	**MSE ↓**	**R^2 ↑**	**Id. Acc. ↑**
EnCodec				
Sliding Window Averaging	0.912	1.254	0.213	0.532
Skipped Timesteps	0.895	1.197	0.236	0.541
Total Averaging	0.874	1.132	0.258	0.559
Chromagram Tokenizer				
Sliding Window Averaging	0.932	1.301	0.192	0.521
Skipped Timesteps	0.917	1.243	0.219	0.529
Total Averaging	0.887	1.178	0.247	0.548
T5 (Text Encoder)				
Sliding Window Averaging	0.832	1.065	0.295	0.574
Skipped Timesteps	0.819	1.011	0.316	0.583
Total Averaging	**0.791**	**0.948**	**0.342**	**0.602**

that the textual descriptions of the music samples provide more discriminative features for predicting the prior embeddings from fMRI data.

These results demonstrate the effectiveness of using a combination of total averaging and the T5 encoder for predicting prior embeddings from fMRI data. However, the overall performance metrics indicate that there is still room for improvement in this task, as the best R^2 score is only 0.342, and the highest identification accuracy is 0.602, which is only slightly better than random chance (0.5). This suggests that predicting prior embeddings from fMRI data is a challenging task that requires further research and development of more sophisticated methods.

6.2 Music Generation Results

Table 6.2 presents the results of music generation using the predicted prior embeddings from different temporal alignment techniques and modality encoders. The generated music samples are compared with the ground truth music using three evaluation metrics: Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD).

The results show that the music generated using the predicted prior embeddings from the EnCodec and Chromagram Tokenizer encoders is of poor quality, as indicated by the high values of FAD, KL, and MCD. This suggests that the predicted embeddings from

Table 6.2: Music generation results using different temporal alignment techniques and modality encoders. Best results for each encoder are in bold.

Method	FAD ↓	KL ↓	MCD ↓
EnCodec			
Sliding Window Averaging	23.76	4.32	8.65
Skipped Timesteps	22.14	4.11	8.27
Total Averaging	20.89	3.95	7.98
Chromagram Tokenizer			
Sliding Window Averaging	25.43	4.56	9.12
Skipped Timesteps	24.09	4.37	8.83
Total Averaging	22.61	4.18	8.51
T5 (Text Encoder)			
Sliding Window Averaging	12.35	2.74	5.39
Skipped Timesteps	11.48	2.59	5.12
Total Averaging	**8.41**	**2.42**	**4.87**

these encoders do not capture the essential characteristics of the music samples, leading to incoherent and low-quality generated music.

On the other hand, the music generated using the predicted prior embeddings from the T5 (text) encoder shows significantly better results, with lower values of FAD, KL, and MCD compared to the other encoders. This is consistent with the findings from the prior embedding prediction task, where the T5 encoder achieved the best performance. Among the different temporal alignment techniques, total averaging yields the best results for the T5 encoder, with an impressive FAD of 8.41, a KL of 2.42, and an MCD of 4.87.

The FAD score of 8.41 for the T5 encoder with total averaging indicates that the generated music has a relatively close distribution of audio features compared to the real music. This suggests that the text-based prior embeddings capture important characteristics of the music samples, enabling the generation of more coherent and realistic music. However, the KL and MCD values still indicate some differences between the generated and ground truth music, implying that there is room for further improvement.

These results highlight the importance of selecting an appropriate modality encoder for generating music from fMRI data. The text-based T5 encoder, combined with the total averaging technique, demonstrates the most promising results among the evaluated methods. This finding suggests that textual descriptions of music samples provide valuable information for reconstructing music from brain activity.

Nevertheless, the overall quality of the generated music still falls short of the ground truth, as evidenced by the non-zero values of the evaluation metrics. Further research is needed to develop more advanced methods for predicting prior embeddings and generating music that more closely resembles the original music samples. Additionally, exploring other modality encoders or combining multiple encoders may lead to further improvements in the quality of the generated music.

6.3 Top-Performing Regions of Interest (ROIs)

In this section, we discuss the top-performing regions of interest (ROIs) identified in our experiments for predicting prior embeddings from fMRI data. Table 6.3 lists the six ROIs that consistently showed the highest correlation scores between the predicted embeddings and the ground truth embeddings across different modality encoders and temporal alignment techniques.

Table 6.3: Top-performing regions of interest (ROIs) for predicting prior embeddings from fMRI data.

Rank	ROI Name	Basic Functions
1	lh_G_temp_sup-Lateral	Auditory processing, language processing, and multimodal integration
2	lh_S_circular_insula_inf	Auditory-motor integration, temporal processing, and emotional processing
3	lh_G_temporal_inf	Visual object recognition, semantic processing, and multimodal integration
4	lh_S_temporal_inf	Auditory-visual integration, language processing, and social cognition
5	lh_G_temporal_middle	Auditory processing, language processing, and semantic memory
6	lh_G_oc-temp_med-Parahip	Episodic memory, spatial navigation, and emotional processing

The identified ROIs are primarily located in the temporal lobe of the left hemisphere, which is known to play a crucial role in auditory processing, language comprehension, and multimodal integration [212, 213]. The superior temporal gyrus (lh_G_temp_sup-Lateral), which ranks first among the top ROIs, is a key region for auditory processing and is involved in the perception and analysis of complex sounds, including music and speech [214, 155]. The middle temporal gyrus (lh_G_temporal_middle) and the inferior temporal gyrus (lh_G_temporal_inf) are also implicated in auditory processing, as well as semantic processing and multimodal integration [215, 216].

The inferior segment of the circular sulcus of the insula (lh_S_circular_insula_inf) and the inferior temporal sulcus (lh_S_temporal_inf) are notable for their roles in auditory-motor integration, temporal processing, and auditory-visual integration [217, 218]. These regions are likely involved in the temporal aspects of music perception and the integration of musical features across different sensory modalities.

Interestingly, the parahippocampal gyrus (lh_G_oc-temp_med-Parahip) also emerges as a top-performing ROI, despite being primarily associated with episodic memory and spatial navigation [219]. However, recent studies have shown that the parahippocampal gyrus is also involved in the emotional processing of music [220, 221], suggesting that emotional aspects of music may play a role in the reconstruction of music from brain activity.

The prominence of these ROIs in predicting prior embeddings from fMRI data provides valuable insights into the neural mechanisms underlying music perception and generation. Our findings suggest that the temporal lobe, particularly the auditory cortex and its associated regions, contains rich information about the musical features and characteristics that can be used to reconstruct music from brain activity. The involvement of regions implicated in multimodal integration and emotional processing highlights the complex and multifaceted nature of music perception and its representation in the brain.

Furthermore, the success of the text-based T5 encoder in generating more coherent and realistic music from predicted prior embeddings suggests that language-related regions, such as the superior and middle temporal gyri, may play a key role in bridging the gap between brain activity and music generation. The semantic and linguistic information captured by the T5 encoder may align well with the neural representations of music in these regions, facilitating the reconstruction of music from brain activity.

These insights can guide future research in the field of music generation from brain scans, focusing on the identified ROIs and their functional roles in music perception and processing. Further investigations into the specific contributions of each ROI and their interactions may lead to the development of more advanced and biologically plausible models for generating music from brain activity. Additionally, exploring the potential of combining information from multiple ROIs and modality encoders may unlock new possibilities for enhancing the quality and diversity of generated music.

In conclusion, our analysis of the top-performing ROIs in predicting prior embeddings from fMRI data sheds light on the neural foundations of music perception and generation, highlighting the importance of auditory, language, and multimodal integration regions in the temporal lobe. These findings contribute to our understanding of the complex interplay between brain activity and musical experience, paving the way for further advancements in the field of music generation from brain scans.

Chapter 7

Future work

The field of music generation from brain scans is a rapidly evolving area with numerous potential avenues for future research. In this chapter, we outline several promising directions that could extend and enhance the work presented in this thesis.

7.1 Extending Analysis to Multiple Subjects

One of the primary limitations of our current study is the focus on a single subject due to computational constraints. To generalize our findings and ensure the robustness of the proposed methodology, it is essential to extend the analysis to a larger cohort of subjects. By incorporating data from multiple individuals, we can account for inter-subject variability in brain activity patterns and musical preferences [222, 223]. This extension would allow us to investigate the consistency of the identified top-performing ROIs across subjects and potentially uncover additional regions that contribute to music generation from brain scans.

7.2 Incorporating CLAP Encoder in MusicGen

Our current experiments utilize the EnCodec, Chromagram Tokenizer, and T5 encoders for generating prior embeddings. However, recent advancements in contrastive language-audio pretraining (CLAP) have shown promising results in capturing the semantic relationships between text and audio [224]. Integrating the CLAP encoder into the MusicGen model could potentially enhance the quality and coherence of the generated music by leveraging the learned associations between textual descriptions and audio features. This integration may also facilitate the generation of music that better aligns with the semantic content of the brain activity, as captured by the text-based prior embeddings.

7.3 Exploring Diffusion Models for Music Generation

Diffusion models have emerged as a powerful framework for generating high-quality images [225, 226] and have recently been adapted for audio generation tasks [227, 228]. These models learn to generate data by reversing a gradual noising process, allowing for the creation of diverse and realistic samples. Exploring the application of diffusion models to music generation from brain scans could potentially yield more naturalistic and expressive musical outputs. By conditioning the diffusion process on the predicted prior embeddings from brain activity, we can guide the generation of music that captures the underlying neural representations of the listening experience.

7.4 Incorporating Temporal Dynamics in Music Generation

Our current approach focuses on generating music based on static prior embeddings derived from fMRI data. However, music is inherently a temporal art form, and the dynamics of brain activity during music listening may contain valuable information for music generation. Incorporating techniques such as recurrent neural networks (RNNs) [229, 230] or transformer-based models [231, 232] could enable the modeling of temporal dependencies in brain activity and music generation. By capturing the evolving patterns of neural responses over time, we can potentially generate music with more complex structures and temporal variations that better reflect the listening experience.

7.5 Investigating Cross-Modal Transfer Learning

Transfer learning has been successfully applied in various domains to leverage knowledge gained from one task or modality to improve performance in another [233, 234]. In the context of music generation from brain scans, exploring cross-modal transfer learning could provide valuable insights and enhance the quality of generated music. For example, we could investigate the transfer of learned representations from visual or linguistic tasks to the music generation task, as there may be shared neural mechanisms underlying the processing of different sensory modalities [235, 236]. By leveraging knowledge from related domains, we can potentially improve the efficiency and effectiveness of music generation from brain scans.

7.6 Conducting Perceptual Evaluations

While objective evaluation metrics such as FAD, KL divergence, and MCD provide quantitative measures of the generated music's quality, it is crucial to also consider the subjective perceptual experiences of listeners. Conducting perceptual evaluations, such as listening tests or user studies, can offer valuable insights into the perceived quality, expressiveness, and emotional impact of the generated music [237, 238]. By gathering feedback from a diverse group of listeners, we can identify strengths and weaknesses of the generated music and guide future improvements in the music generation pipeline.

7.7 Exploring Brain-Computer Interfaces for Music Generation

The ultimate goal of music generation from brain scans is to create a seamless and intuitive interface between the human mind and musical creativity. Brain-computer interfaces (BCIs) have shown promise in enabling direct communication between the brain and external devices [239, 240]. Integrating our music generation framework with BCI technology could open up new possibilities for real-time, thought-controlled music composition and performance. By developing robust and reliable BCI systems that can accurately decode brain activity and translate it into musical parameters, we can empower individuals to create music using only their thoughts, revolutionizing the way we interact with and express ourselves through music.

In conclusion, the future of music generation from brain scans holds immense potential for both scientific understanding and creative applications. By extending our analysis to multiple subjects, incorporating advanced encoders and generative models, exploring temporal dynamics and cross-modal transfer learning, conducting perceptual evaluations, and integrating with brain-computer interfaces, we can push the boundaries of this exciting field. Through interdisciplinary collaborations and continuous innovation, we can unlock new possibilities for understanding the neural basis of musical creativity and enabling individuals to express themselves through the power of thought and sound.

Chapter 8

Conclusion

In this work, we have explored the fascinating intersection of neuroscience, artificial intelligence, and music generation. Our primary objective was to investigate the feasibility of reconstructing music from brain activity recorded through functional magnetic resonance imaging (fMRI) scans. By leveraging advanced machine learning techniques and state-of-the-art music generation models, we have developed a novel framework that bridges the gap between neural responses to music and the creative process of music composition.

Our research journey began with a comprehensive literature review, where we delved into the existing body of knowledge on the neural correlates of music perception, the principles of brain decoding, and the latest advancements in music generation algorithms. This extensive exploration laid the foundation for our innovative approach to music generation from brain scans.

The core of our methodology revolved around the Map method, a technique that learns a mapping between the fMRI response tensor and the prior embedding space of a conditional music generation model. We experimented with various modality encoders, including EnCodec for audio waveforms, Chromagram Tokenizer for melodic representations, and T5 for textual descriptions of music. Additionally, we tackled the temporal alignment problem by investigating different techniques such as sliding window averaging, skipped timesteps, and total averaging.

Our experiments yielded promising results, demonstrating the potential of generating music from brain activity. The T5 encoder, which leverages textual descriptions of music, emerged as the most effective modality for capturing the semantic and emotional aspects of music perception in the brain. The total averaging technique for temporal alignment also proved to be the most successful in aligning the fMRI response tensor with the prior embedding space.

Furthermore, our analysis of the top-performing regions of interest (ROIs) in the brain provided valuable insights into the neural mechanisms underlying music perception and generation. The identified ROIs, primarily located in the temporal lobe and associated with auditory processing, language comprehension, and multimodal integration, highlight the complex interplay between different brain regions in the experience of music.

The evaluation of the generated music samples using objective metrics such as Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and Mel Cepstral Distortion (MCD) revealed the challenges and limitations of our current approach. While the T5 encoder with total averaging achieved the best performance among the evaluated methods, there is still room for improvement in terms of the quality and fidelity of the generated music compared to the ground truth.

Despite these challenges, our work has laid the groundwork for further advancements in the field of music generation from brain scans. The proposed framework and the insights gained from our experiments open up numerous avenues for future research. Extending the analysis to multiple subjects, incorporating advanced encoders and generative models, exploring temporal dynamics and cross-modal transfer learning, conducting perceptual evaluations, and integrating with brain-computer interfaces are just a few of the exciting directions that can be pursued to push the boundaries of this interdisciplinary field.

The implications of our research extend beyond the realm of music generation. By unravelling the neural codes of music perception and translating them into creative expressions, we contribute to a deeper understanding of the human brain and its relationship with the arts. Our work has the potential to inspire new forms of musical creation, enable individuals with limited motor abilities to compose music using their thoughts, and foster collaborations between neuroscientists, musicians, and AI researchers.

In conclusion, this thesis represents a significant step forward in the quest to generate music from brain scans. Through our innovative methodology, comprehensive experiments, and insightful analysis, we have demonstrated the feasibility and potential of this exciting research direction. As we continue to unravel the mysteries of the brain and harness the power of artificial intelligence, we move closer to a future where the boundaries between the mind, music, and machine dissolve, giving rise to new forms of creative expression and artistic exploration.

Bibliography

[1] Y.-H. Chin, C.-H. Lin, E. Siahaan, J.-C. Wang, *et al.*, “Music emotion detection using hierarchical sparse kernel machines,” *The Scientific World Journal*, vol. 2014, 2014.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 5998–6008, 2017.

[3] S. J. Prince, *Understanding Deep Learning.* MIT Press, 2023.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.

[6] Z. Li, H. Wang, and X. Jiang, “Audioformer: Audio transformer learns audio feature representations from discrete acoustic codes,” *arXiv preprint arXiv:2308.07221*, 2023.

[7] C. Destrieux, B. Fischl, A. Dale, and E. Halgren, “Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature,” *Neuroimage*, vol. 53, no. 1, pp. 1–15, 2010.

[8] R. Rhoades and R. G. Pflanzer, “Human physiology,” *(No Title)*, 2003.

[9] L. S. Hamilton, Y. Oganian, J. Hall, and E. F. Chang, “Parallel and distributed encoding of speech across human auditory cortex,” *Cell*, vol. 184, no. 18, pp. 4626–4639, 2021.

[10] H. R. Prakash, M. Korostenskaja, K. Lee, J. Baumgartner, E. Castillo, and U. Bagci, "Automatic response assessment in regions of language cortex in epilepsy patients using ecog-based functional mapping and machine learning," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 519–524, IEEE, 2017.

[11] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] O. Esteban, C. J. Markiewicz, R. W. Blair, C. A. Moodie, A. I. Isik, A. Erramuzpe, J. D. Kent, M. Goncalves, E. DuPre, M. Snyder, *et al.*, "fmriprep: a robust preprocessing pipeline for functional mri," *Nature methods*, vol. 16, no. 1, pp. 111–116, 2019.

[13] V. Alluri, P. Toiviainen, I. P. Jääskeläinen, E. Glerean, M. Sams, and E. Brattico, "Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm," *NeuroImage*, vol. 59, no. 4, pp. 3677–3689, 2012.

[14] P. Toiviainen, V. Alluri, E. Brattico, M. Wallentin, and P. Vuust, "Capturing the musical brain with lasso: Dynamic decoding of musical features from fmri data," *NeuroImage*, vol. 88, pp. 170–180, 2014.

[15] E. J. Allen, M. Moerel, A. Lage-Castellanos, F. De Martino, E. Formisano, and A. J. Oxenham, "Encoding of natural timbre dimensions in human auditory cortex," *NeuroImage*, vol. 166, pp. 60–70, 2018.

[16] S. Koelsch, T. Fritz, D. Y. v. Cramon, K. Müller, and A. D. Friederici, "Investigating emotion with music: An fmri study," *Human Brain Mapping*, vol. 27, no. 3, pp. 239–250, 2006.

[17] M. A. Casey, "Music of the 7ts: Predicting and decoding multivoxel fmri responses with acoustic, schematic, and categorical music features," *Frontiers in psychology*, vol. 8, p. 1179, 2017.

[18] T. Nakai, N. Koide-Majima, and S. Nishimoto, "Correspondence of categorical and feature-based representations of music in the human brain," *Brain and Behavior*, vol. 11, no. 1, p. e01936, 2021.

[19] S. Forsgren and H. Martiros, "Riffusion - Stable diffusion for real-time music generation," 2022.

[20] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[21] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, "Noise2music: Text-conditioned music generation with diffusion models," 2023.

[22] M. W. Y. Lam, Q. Tian, T. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song, J. Chen, Y. Wang, and Y. Wang, "Efficient neural music generation," 2023.

[23] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.

[24] A. Degerman, T. Rinne, J. Pekkola, T. Autti, I. P. Jääskeläinen, M. Sams, and K. Alho, "Human brain activity associated with audiovisual perception and attention," *Neuroimage*, vol. 34, no. 4, pp. 1683–1691, 2007.

[25] C. D. Gilbert and M. Sigman, "Brain states: top-down influences in sensory processing," *Neuron*, vol. 54, no. 5, pp. 677–696, 2007.

[26] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann, "Neurophysiological investigation of the basis of the fmri signal," *nature*, vol. 412, no. 6843, pp. 150–157, 2001.

[27] M. Meyer, S. Baumann, and L. Jancke, "Electrical brain imaging reveals spatiotemporal dynamics of timbre perception in humans," *Neuroimage*, vol. 32, no. 4, pp. 1510–1523, 2006.

[28] G. Ganis, W. L. Thompson, and S. M. Kosslyn, "Brain areas underlying visual mental imagery and visual perception: an fmri study," *Cognitive Brain Research*, vol. 20, no. 2, pp. 226–241, 2004.

[29] J. Lim, D. Lin, W. J. Sohn, C. M. McCrimmon, P. T. Wang, Z. Nenadic, and A. H. Do, "Bci-based neuroprostheses and physiotherapies for stroke motor rehabilitation," in *Neurorehabilitation Technology*, pp. 509–524, Springer, 2022.

[30] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, "Eeg-based neuroprosthesis control: a step towards clinical practice," *Neuroscience letters*, vol. 382, no. 1-2, pp. 169–174, 2005.

[31] D. Wen, B. Liang, Y. Zhou, H. Chen, and T.-P. Jung, "The current research of combining multi-modal brain-computer interfaces with virtual reality," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 9, pp. 3278–3287, 2020.

[32] A. Ravi, J. Lu, S. Pearce, and N. Jiang, "Enhanced system robustness of asynchronous bci in augmented reality using steady-state motion visual evoked potential," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 85–95, 2022.

[33] C.-Y. Chiu, A. K. Singh, Y.-K. Wang, J.-T. King, and C.-T. Lin, "A wireless steady state visually evoked potential-based bci eating assistive system," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3003–3007, IEEE, 2017.

[34] X. Chen, B. Zhao, Y. Wang, and X. Gao, "Combination of high-frequency ssvep-based bci and computer vision for controlling a robotic arm," *Journal of neural engineering*, vol. 16, no. 2, p. 026012, 2019.

[35] M. X. Cohen, "Where does eeg come from and what does it mean?," *Trends in neurosciences*, vol. 40, no. 4, pp. 208–218, 2017.

[36] D. J. Heeger and D. Ress, "What does fmri tell us about neuronal activity?," *Nature reviews neuroscience*, vol. 3, no. 2, pp. 142–151, 2002.

[37] F. L. da Silva, "Eeg and meg: relevance to neuroscience," *Neuron*, vol. 80, no. 5, pp. 1112–1128, 2013.

[38] K. J. Friston, "Models of brain function in neuroimaging," *Annu. Rev. Psychol.*, vol. 56, pp. 57–87, 2005.

[39] J. D. Power, D. A. Fair, B. L. Schlaggar, and S. E. Petersen, "The development of human functional brain networks," *Neuron*, vol. 67, no. 5, pp. 735–748, 2010.

[40] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the national academy of sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[41] A. J. Kell, D. L. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy," *Neuron*, vol. 98, pp. 630–644.e16, May 2018.

[42] U. Güçlü and M. A. van Gerven, “Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream,” *Journal of Neuroscience*, vol. 35, no. 27, pp. 10005–10014, 2015.

[43] L. Cross, J. Cockburn, Y. Yue, and J. P. O’Doherty, “Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments,” *Neuron*, vol. 109, pp. 724–738.e7, Feb. 2021.

[44] Y. Takagi and S. Nishimoto, “High-resolution image reconstruction with latent diffusion models from human brain activity,” *bioRxiv*, 2022.

[45] G. Shen, T. Horikawa, K. Majima, and Y. Kamitani, “Deep image reconstruction from human brain activity,” *PLOS Computational Biology*, vol. 15, p. e1006633, Jan. 2019.

[46] Z. Chen, J. Qing, T. Xiang, W. L. Yue, and J. H. Zhou, “Seeing beyond the brain: Masked modeling conditioned diffusion model for human vision decoding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[47] Y. Takagi and S. Nishimoto, “Improving visual image reconstruction from human brain activity using latent diffusion models via multiple decoded inputs,” 2023.

[48] R. Santoro, M. Moerel, F. D. Martino, G. Valente, K. Ugurbil, E. Yacoub, and E. Formisano, “Reconstructing the spectrotemporal modulations of real-life sounds from fMRI response patterns,” *Proceedings of the National Academy of Sciences*, vol. 114, pp. 4799–4804, Apr. 2017.

[49] J.-Y. Park, M. Tsukamoto, M. Tanaka, and Y. Kamitani, “Sound reconstruction from human brain activity via a generative model with brain-like auditory features,” 2023.

[50] T. Nakai, N. Koide-Majima, and S. Nishimoto, “Music genre neuroimaging dataset,” *Data in Brief*, vol. 40, p. 107675, 2022.

[51] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, *et al.*, “On generative spoken language modeling from raw audio,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[52] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[53] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[54] S. Bassan, Y. Adi, and J. S. Rosenschein, "Unsupervised symbolic music segmentation using ensemble temporal prediction errors," *arXiv preprint arXiv:2207.00760*, 2022.

[55] A. Ycart, E. Benetos, *et al.*, "A study on lstm networks for polyphonic music sequence modelling," ISMIR, 2017.

[56] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv preprint arXiv:2011.06801*, 2020.

[57] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[58] C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba, "Foley music: Learning to generate music from videos," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 758–775, Springer, 2020.

[59] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[60] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, "Mulan: A joint embedding of music audio and natural language," *arXiv preprint arXiv:2208.12415*, 2022.

[61] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, *et al.*, "Singsong: Generating musical accompaniments from singing," *arXiv preprint arXiv:2301.12662*, 2023.

[62] F. Schneider, Z. Jin, and B. Schölkopf, "Mo\ˆ usai: Text-to-music generation with long-context latent diffusion," *arXiv preprint arXiv:2301.11757*, 2023.

[63] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, *et al.*, "Noise2music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.

[64] K. Maina, "Msanii: High fidelity music synthesis on a shoestring budget," *arXiv preprint arXiv:2301.06468*, 2023.

[65] S. Forsgren and H. Martiros, "Riffusion-stable diffusion for real-time music generation. 2022," *URL https://riffusion. com/about.*

[66] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[67] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *arXiv preprint arXiv:2207.09983*, 2022.

[68] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[69] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[70] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[71] R. Sheffer and Y. Adi, "I hear your true colors: Image guided audio generation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.

[72] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," *arXiv preprint arXiv:2301.12661*, 2023.

[73] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[74] F. Tong, “Primary visual cortex and visual awareness,” *Nature Reviews Neuroscience*, vol. 4, no. 3, pp. 219–229, 2003.

[75] K. Grill-Spector and R. Malach, “The human visual cortex,” *Annu. Rev. Neurosci.*, vol. 27, pp. 649–677, 2004.

[76] E. J. Allen, G. St-Yves, Y. Wu, J. L. Breedlove, J. S. Prince, L. T. Dowdle, M. Nau, B. Caron, F. Pestilli, I. Charest, *et al.*, “A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence,” *Nature neuroscience*, vol. 25, no. 1, pp. 116–126, 2022.

[77] T. Horikawa and Y. Kamitani, “Generic decoding of seen and imagined objects using hierarchical visual features,” *Nature communications*, vol. 8, no. 1, p. 15037, 2017.

[78] G. Shen, T. Horikawa, K. Majima, and Y. Kamitani, “Deep image reconstruction from human brain activity,” *PLoS computational biology*, vol. 15, no. 1, p. e1006633, 2019.

[79] Y. Takagi and S. Nishimoto, “High-resolution image reconstruction with latent diffusion models from human brain activity,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14453–14463, 2023.

[80] F. Ozcelik and R. VanRullen, “Natural scene reconstruction from fmri signals using generative latent diffusion,” *Scientific Reports*, vol. 13, no. 1, p. 15666, 2023.

[81] T. Fang, Y. Qi, and G. Pan, “Reconstructing perceptive images from brain activity by shape-semantic gan,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13038–13048, 2020.

[82] Z. Ren, J. Li, X. Xue, X. Li, F. Yang, Z. Jiao, and X. Gao, “Reconstructing seen image from brain activity by visually-guided cognitive representation and adversarial learning,” *NeuroImage*, vol. 228, p. 117602, 2021.

[83] R. Beliy, G. Gaziv, A. Hoogi, F. Strappini, T. Golan, and M. Irani, “From voxels to pixels and back: Self-supervision in natural-image reconstruction from fmri,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[84] G. Gaziv, R. Beliy, N. Granot, A. Hoogi, F. Strappini, T. Golan, and M. Irani, “Self-supervised natural image reconstruction and large-scale semantic classification from brain activity,” *NeuroImage*, vol. 254, p. 119121, 2022.

[85] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, and M. Shah, "Deep learning human mind for automated visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6809–6817, 2017.

[86] Y. Bai, X. Wang, Y. Cao, Y. Ge, C. Yuan, and Y. Shan, "Dreamdiffusion: Generating high-quality images from brain eeg signals," *arXiv preprint arXiv:2306.16934*, 2023.

[87] Y.-T. Lan, K. Ren, Y. Wang, W.-L. Zheng, D. Li, B.-L. Lu, and L. Qiu, "Seeing through the brain: Image reconstruction of visual perception from human brain signals," *arXiv preprint arXiv:2308.02510*, 2023.

[88] J.-Y. Park, M. Tsukamoto, M. Tanaka, and Y. Kamitani, "Sound reconstruction from human brain activity via a generative model with brain-like auditory features," *arXiv preprint arXiv:2306.11629*, 2023.

[89] S. A. Nastase, Y.-F. Liu, H. Hillman, A. Zadbood, L. Hasenfratz, N. Keshavarzian, J. Chen, C. J. Honey, Y. Yeshurun, M. Regev, *et al.*, "The "narratives" fmri dataset for evaluating models of naturalistic language comprehension," *Scientific data*, vol. 8, no. 1, p. 250, 2021.

[90] N. Xi, S. Zhao, H. Wang, C. Liu, B. Qin, and T. Liu, "Unicorn: Unified cognitive signal reconstruction bridging cognitive signals and human language," *arXiv preprint arXiv:2307.05355*, 2023.

[91] Y. Guo, T. Liu, X. Zhang, A. Wang, and W. Wang, "End-to-end translation of human neural activity to speech with a dual–dual generative adversarial network," *Knowledge-Based Systems*, vol. 277, p. 110837, 2023.

[92] J. Tang, A. LeBel, S. Jain, and A. G. Huth, "Semantic reconstruction of continuous language from non-invasive brain recordings," *Nature Neuroscience*, pp. 1–9, 2023.

[93] T. Nakai, N. Koide-Majima, and S. Nishimoto, "Music genre neuroimaging dataset," *Data in Brief*, vol. 40, p. 107675, 2022.

[94] I. Daly, N. Nicolaou, D. Williams, F. Hwang, A. Kirke, E. Miranda, and S. J. Nasuto, "Neural and physiological data from participants listening to affective music," *Scientific Data*, vol. 7, no. 1, p. 177, 2020.

[95] T. I. Denk, Y. Takagi, T. Matsuyama, A. Agostinelli, T. Nakai, C. Frank, and S. Nishimoto, "Brain2music: Reconstructing music from human brain activity," *arXiv preprint arXiv:2307.11078*, 2023.

[96] I. Daly, "Neural decoding of music from the eeg," *Scientific Reports*, vol. 13, no. 1, p. 624, 2023.

[97] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[98] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[99] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *International conference on machine learning*, pp. 3918–3926, PMLR, 2018.

[100] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *arXiv preprint arXiv:1909.11646*, 2019.

[101] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, "Flowavenet: A generative flow for raw audio," *arXiv preprint arXiv:1811.02155*, 2018.

[102] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[103] R. Yamamoto, E. Song, and J.-M. Kim, "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation," *arXiv preprint arXiv:1904.04472*, 2019.

[104] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*, pp. 2410–2419, PMLR, 2018.

[105] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[106] P. Neekhara, C. Donahue, M. Puckette, S. Dubnov, and J. McAuley, "Expediting tts synthesis with adversarial vocoding," *arXiv preprint arXiv:1904.07944*, 2019.

[107] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 94–98, 2018.

[108] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621, IEEE, 2019.

[109] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *International conference on machine learning*, pp. 7586–7598, PMLR, 2020.

[110] C. Aouameur, P. Esling, and G. Hadjeres, "Neural drum machine: An interactive system for real-time synthesis of drum sounds," *arXiv preprint arXiv:1907.02637*, 2019.

[111] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in neural information processing systems*, vol. 32, 2019.

[112] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[113] R. Liu, B. Sisman, F. Bao, G. Gao, and H. Li, "Wavetts: Tacotron-based tts with joint time-frequency domain loss," *arXiv preprint arXiv:2002.00417*, 2020.

[114] S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," *arXiv preprint arXiv:1906.01083*, 2019.

[115] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet (cyclegan (cqt (audio))) pipeline for musical timbre transfer," *arXiv preprint arXiv:1811.09620*, 2018.

[116] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-q transform with non-stationary gabor frames," *Proceedings of DAFX11, Paris*, vol. 33, p. 81, 2011.

[117] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*, pp. 1068–1077, PMLR, 2017.

[118] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[119] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[120] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[121] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International conference on machine learning*, pp. 5180–5189, PMLR, 2018.

[122] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.

[123] A. Défossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, "Sing: Symbol-to-instrument neural generator," *Advances in neural information processing systems*, vol. 31, 2018.

[124] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5891–5895, IEEE, 2019.

[125] K. Subramani, P. Rao, and A. D'Hooge, "Vapar synth-a variational parametric model for audio synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 796–800, IEEE, 2020.

[126] L. Juvela, B. Bollepalli, V. Tsiaras, and P. Alku, "Glotnet—a raw waveform model for the glottal excitation in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 1019–1030, 2019.

[127] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," *arXiv preprint arXiv:1704.03809*, 2017.

[128] S. Yang, L. Xie, X. Chen, X. Lou, X. Zhu, D. Huang, and H. Li, "Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework," in *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 685–691, IEEE, 2017.

[129] G. E. Henter, J. Lorenzo-Trueba, X. Wang, and J. Yamagishi, "Deep encoder-decoder models for unsupervised learning of controllable speech synthesis," *arXiv preprint arXiv:1807.11470*, 2018.

[130] A. Bitton, P. Esling, and T. Harada, "Neural granular sound synthesis," *arXiv preprint arXiv:2008.01393*, 2020.

[131] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[132] P. Esling, A. Bitton, *et al.*, "Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics," *arXiv preprint arXiv:1805.08501*, 2018.

[133] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Flow synthesizer: Universal audio synthesizer control with normalizing flows," *Applied Sciences*, vol. 10, no. 1, p. 302, 2019.

[134] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," *arXiv preprint arXiv:1703.10847*, 2017.

[135] H. H. Mao, T. Shin, and G. Cottrell, "Deepj: Style-specific music generation," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 377–382, IEEE, 2018.

[136] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[137] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[138] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[139] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[140] A. Huang, M. Dinculescu, A. Vaswani, and D. Eck, "Visualizing music self-attention," in *Proc. NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language*, vol. 1, p. 4, 2018.

[141] N. Shazeer, Y. Cheng, N. Parmar, D. Tran, A. Vaswani, P. Koanantakool, P. Hawkins, H. Lee, M. Hong, C. Young, *et al.*, "Mesh-tensorflow: Deep learning for supercomputers," *Advances in neural information processing systems*, vol. 31, 2018.

[142] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

[143] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[144] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.

[145] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[146] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17022–17033, 2020.

[147] M. Tagliasacchi, Y. Li, K. Misiunas, and D. Roblek, "Seanet: A multi-modal speech enhancement network," *arXiv preprint arXiv:2009.02095*, 2020.

[148] Y. Li, M. Tagliasacchi, O. Rybakov, V. Ungureanu, and D. Roblek, "Real-time speech frequency bandwidth extension," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 691–695, IEEE, 2021.

[149] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[150] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, S. Mack, *et al.*, *Principles of neural science*, vol. 4. McGraw-hill New York, 2000.

[151] D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, J. O. McNamara, and S. M. Williams, *Neuroscience*. Sunderland, MA: Sinauer Associates, 2001.

[152] R. J. Zatorre, J. L. Chen, and V. B. Penhune, "When the brain plays music: auditory–motor interactions in music perception and production," *Nature reviews neuroscience*, vol. 8, no. 7, pp. 547–558, 2007.

[153] E. Formisano, D.-S. Kim, F. Di Salle, P.-F. van de Moortele, K. Ugurbil, and R. Goebel, "Mirror-symmetric tonotopic maps in human primary auditory cortex," *Neuron*, vol. 40, no. 4, pp. 859–869, 2003.

[154] T. D. Griffiths, G. Rees, A. Rees, G. G. Green, C. Witton, D. Rowe, C. Büchel, R. Turner, and R. S. Frackowiak, "Human brain areas involved in the analysis of auditory movement," *Human brain mapping*, vol. 7, no. 4, pp. 290–300, 1999.

[155] A. M. Leaver and J. P. Rauschecker, "Cortical representation of perceived musical rhythm," *Journal of Neuroscience*, vol. 30, no. 2, pp. 426–434, 2010.

[156] E. K. Miller and J. D. Cohen, "The prefrontal cortex and cognitive control," *Nature reviews neuroscience*, vol. 1, no. 1, pp. 59–65, 2000.

[157] P. Janata, J. L. Birk, J. D. Van Horn, M. Leman, B. Tillmann, and J. J. Bharucha, "The cortical topography of tonal structures underlying western music," *science*, vol. 298, no. 5601, pp. 2167–2170, 2002.

[158] S. Koelsch, T. C. Gunter, M. Wittfoth, and D. Sammler, "Interaction between syntax processing in language and in music: An erp study," *Journal of cognitive neuroscience*, vol. 17, no. 10, pp. 1565–1577, 2005.

[159] B. Maess, S. Koelsch, T. C. Gunter, and A. D. Friederici, "Musical syntax is processed in broca's area: an meg study," *Nature neuroscience*, vol. 4, no. 5, pp. 540–545, 2001.

[160] S. Koelsch, T. Fritz, D. Y. v. Cramon, K. Müller, and A. D. Friederici, "Investigating emotion with music: An fmri study," *Human brain mapping*, vol. 27, no. 3, pp. 239–250, 2005.

[161] J. A. Grahn and J. D. McAuley, "Neural bases of individual differences in beat perception," *NeuroImage*, vol. 47, no. 4, pp. 1894–1903, 2009.

[162] O. Sporns, "Cerebral cartography and connectomics," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1668, p. 20140173, 2015.

[163] M. J. McKeown, S. Makeig, G. G. Brown, T.-P. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski, "Analysis of fmri data by blind separation into independent spatial components," *Human brain mapping*, vol. 6, no. 3, pp. 160–188, 1998.

[164] W. Mai, J. Zhang, P. Fang, and Z. Zhang, "Brain-conditional multimodal synthesis: A survey and taxonomy," *arXiv preprint arXiv:2401.00430*, 2023.

[165] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in mri data," *IEEE transactions on medical imaging*, vol. 17, no. 1, pp. 87–97, 1998.

[166] A. M. Dale, B. Fischl, and M. I. Sereno, "Cortical surface-based analysis: I. segmentation and surface reconstruction," *Neuroimage*, vol. 9, no. 2, pp. 179–194, 1999.

[167] B. Fischl, M. I. Sereno, and A. M. Dale, "Cortical surface-based analysis ii: Inflation, flattening, and a surface-based coordinate system," *Neuroimage*, vol. 9, no. 2, pp. 195–207, 1999.

[168] D. N. Greve and B. Fischl, "Accurate and robust brain image alignment using boundary-based registration," *Neuroimage*, vol. 48, no. 1, pp. 63–72, 2009.

[169] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T. A. Nguyen, M. Riviere, A. Mohamed, E. Dupoux, *et al.*, "Text-free prosody-aware generative spoken language modeling," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8666–8681, 2022.

[170] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[171] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[172] Y. Wu*, K. Chen*, T. Zhang*, Y. Hui*, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[173] Y. Takagi and S. Nishimoto, "Improving visual image reconstruction from human brain activity using latent diffusion models via multiple decoded inputs," *arXiv preprint arXiv:2306.11536*, 2023.

[174] W. Mai and Z. Zhang, "Unibrain: Unify image reconstruction and captioning all in one diffusion model from human brain activity," *arXiv preprint arXiv:2308.07428*, 2023.

[175] Y. Lu, C. Du, D. Wang, and H. He, "Minddiffuser: Controlled image reconstruction from human brain activity with semantic and structural diffusion," *arXiv preprint arXiv:2303.14139*, 2023.

[176] P. S. Scotti, A. Banerjee, J. Goode, S. Shabalin, A. Nguyen, E. Cohen, A. J. Dempster, N. Verlinde, E. Yundler, D. Weisberg, *et al.*, "Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors," *arXiv preprint arXiv:2305.18274*, 2023.

[177] Y. Liu, Y. Ma, W. Zhou, G. Zhu, and N. Zheng, "Brainclip: Bridging brain and visual-linguistic representation via clip for generic natural visual stimulus decoding from fmri," *arXiv preprint arXiv:2302.12971*, 2023.

[178] M. Ferrante, F. Ozcelik, T. Boccato, R. VanRullen, and N. Toschi, "Brain captioning: Decoding human brain activity into images and text," *arXiv preprint arXiv:2305.11560*, 2023.

[179] M. Ferrante, T. Boccato, and N. Toschi, "Semantic brain decoding: from fmri to conceptually similar image reconstruction of visual stimuli," *arXiv preprint arXiv:2212.06726*, 2022.

[180] Z. Gu, K. Jamison, A. Kuceyeski, and M. Sabuncu, "Decoding natural image stimuli from fmri data with a surface-based convolutional network," *arXiv preprint arXiv:2212.02409*, 2022.

[181] S. Lin, T. Sprague, and A. K. Singh, "Mind reader: Reconstructing complex images from brain activities," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29624–29636, 2022.

[182] F. Ozcelik, B. Choksi, M. Mozafari, L. Reddy, and R. VanRullen, "Reconstruction of perceived images from fmri patterns and semantic brain exploration using instance-conditioned gans," in *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2022.

[183] L. Meng and C. Yang, "Semantics-guided hierarchical feature encoding generative adversarial network for natural image reconstruction from brain activities," in *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2023.

[184] M. Mozafari, L. Reddy, and R. VanRullen, "Reconstructing natural scenes from fmri patterns using bigbigan," in *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[185] K. Seeliger, U. Güçlü, L. Ambrogioni, Y. Güçlütürk, and M. A. van Gerven, "Generative adversarial networks for reconstructing natural images from brain activity," *NeuroImage*, vol. 181, pp. 775–785, 2018.

[186] K. Qiao, J. Chen, L. Wang, C. Zhang, L. Tong, and B. Yan, "Biggan-based bayesian reconstruction of natural images from human brain activity," *Neuroscience*, vol. 444, pp. 92–105, 2020.

[187] R. VanRullen and L. Reddy, "Reconstructing faces from fmri patterns using deep generative neural networks," *Communications biology*, vol. 2, no. 1, p. 193, 2019.

[188] J. Chen, Y. Qi, and G. Pan, "Rethinking visual reconstruction: Experience-based content completion guided by visual cues," in *Proceedings of the 40th International Conference on Machine Learning* (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of *Proceedings of Machine Learning Research*, pp. 4856–4866, PMLR, 23–29 Jul 2023.

[189] E. Miliotou, P. Kyriakis, J. D. Hinman, A. Irimia, and P. Bogdan, "Generative decoding of visual stimuli," in *Proceedings of the 40th International Conference on Machine Learning* (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of *Proceedings of Machine Learning Research*, pp. 24775–24784, PMLR, 23–29 Jul 2023.

[190] L. Meng and C. Yang, "Dual-guided brain diffusion model: Natural image reconstruction from human visual stimulus fmri," *Bioengineering*, vol. 10, no. 10, p. 1117, 2023.

[191] W. Xia, R. de Charette, C. Öztireli, and J.-H. Xue, "Dream: Visual decoding from reversing human visual system," *arXiv preprint arXiv:2310.02265*, 2023.

[192] Y. Benchetrit, H. Banville, and J.-R. King, "Brain decoding: toward real-time reconstruction of visual perception," *arXiv preprint arXiv:2310.19812*, 2023.

[193] K. Han, H. Wen, J. Shi, K.-H. Lu, Y. Zhang, D. Fu, and Z. Liu, "Variational autoencoder: An unsupervised model for encoding and decoding fmri activity in visual cortex," *NeuroImage*, vol. 198, pp. 125–136, 2019.

[194] S. Chatterjee and D. Samanta, "Dreamcatcher: Revealing the language of the brain with fmri using gpt embedding," *arXiv preprint arXiv:2306.10082*, 2023.

[195] S. Takada, R. Togo, T. Ogawa, and M. Haseyama, "Generation of viewed image captions from human brain activity via unsupervised text latent space," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2521–2525, IEEE, 2020.

[196] E. Matsuo, I. Kobayashi, S. Nishimoto, S. Nishida, and H. Asoh, "Describing semantic representations of brain activity evoked by visual stimuli," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 576–583, IEEE, 2018.

[197] E. Matsuo, I. Kobayashi, S. Nishimoto, S. Nishida, and H. Asoh, "Generating natural language descriptions for semantic representations of human brain activity," in *Proceedings of the ACL 2016 student research workshop*, pp. 22–29, 2016.

[198] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[199] S. Norman-Haignere, N. G. Kanwisher, and J. H. McDermott, "Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition," *neuron*, vol. 88, no. 6, pp. 1281–1296, 2015.

[200] S. Moeller, E. Yacoub, C. A. Olman, E. Auerbach, J. Strupp, N. Harel, and K. Uğurbil, "Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri," *Magnetic resonance in medicine*, vol. 63, no. 5, pp. 1144–1153, 2010.

[201] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[202] M. Jenkinson, P. Bannister, M. Brady, and S. Smith, "Improved optimization for the robust and accurate linear registration and motion correction of brain images," *Neuroimage*, vol. 17, no. 2, pp. 825–841, 2002.

[203] T. D. la Tour, M. Eickenberg, A. O. Nunez-Elizalde, and J. L. Gallant, "Feature-space selection with banded ridge regression," *NeuroImage*, vol. 264, p. 119728, 2022.

[204] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[205] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.

[206] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

[207] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, IEEE, 2017.

[208] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2014.

[209] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[210] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780, IEEE, 2017.

[211] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, pp. 125–128, IEEE, 1993.

[212] G. Hickok and D. Poeppel, "The cortical organization of speech processing," *Nature Reviews Neuroscience*, vol. 8, no. 5, pp. 393–402, 2007.

[213] J. R. Binder, R. H. Desai, W. W. Graves, and L. L. Conant, "Where is the semantic system? a critical review and meta-analysis of 120 functional neuroimaging studies," *Cerebral Cortex*, vol. 19, no. 12, pp. 2767–2796, 2009.

[214] R. J. Zatorre, P. Belin, and V. B. Penhune, "Structure and function of auditory cortex: music and speech," *Trends in Cognitive Sciences*, vol. 6, no. 1, pp. 37–46, 2002.

[215] M. Visser, E. Jefferies, K. V. Embleton, and M. A. Lambon Ralph, "Both the middle temporal gyrus and the ventral anterior temporal area are crucial for multimodal

semantic processing," *Journal of Cognitive Neuroscience*, vol. 24, no. 8, pp. 1766–1778, 2012.

[216] R. J. Binney, M. L. Henry, M. Babiak, P. S. Pressman, H. J. Rosen, B. L. Miller, and K. P. Rankin, "The convergent and divergent validity of the ruff figural fluency test," *Neuropsychological Assessment*, vol. 7, no. 1, pp. 33–45, 2010.

[217] D.-E. Bamiou, F. E. Musiek, and L. M. Luxon, "The insula (island of reil) and its role in auditory processing: literature review," *Brain Research Reviews*, vol. 42, no. 2, pp. 143–154, 2003.

[218] M. S. Beauchamp, B. D. Argall, J. Bodurka, J. H. Duyn, and A. Martin, "Unraveling multisensory integration: patchy organization within human sts multisensory cortex," *Nature Neuroscience*, vol. 7, no. 11, pp. 1190–1192, 2004.

[219] E. M. Aminoff, K. Kveraga, and M. Bar, "The role of the parahippocampal cortex in cognition," *Trends in Cognitive Sciences*, vol. 17, no. 8, pp. 379–390, 2013.

[220] S. Koelsch, "Brain correlates of music-evoked emotions," *Nature Reviews Neuroscience*, vol. 15, no. 3, pp. 170–180, 2014.

[221] S. Frühholz, W. Trost, and D. Grandjean, "Functional decoding and meta-analytic connectivity modeling of the human medial geniculate body in fmri," *NeuroImage*, vol. 96, pp. 95–106, 2014.

[222] E. S. Finn, X. Shen, D. Scheinost, M. D. Rosenberg, J. Huang, M. M. Chun, X. Papademetris, and R. T. Constable, "Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity," *Nature Neuroscience*, vol. 18, no. 11, pp. 1664–1671, 2015.

[223] R. W. Wilkins, D. A. Hodges, P. J. Laurienti, M. Steen, and J. H. Burdette, "Network science and the effects of music preference on functional brain connectivity: from beethoven to eminem," *Scientific Reports*, vol. 4, no. 1, pp. 1–7, 2014.

[224] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[225] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020.

[226] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," *arXiv preprint arXiv:2102.09672*, 2021.

[227] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2020.

[228] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[229] D. Eck and J. Schmidhuber, "Finding temporal structure in music: Blues improvisation with lstm recurrent networks," in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 747–756, IEEE, 2002.

[230] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," *arXiv preprint arXiv:1604.08723*, 2016.

[231] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *International Conference on Learning Representations*, 2018.

[232] C. Donahue, I. Simon, and S. Dieleman, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pp. 685–692, 2019.

[233] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[234] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, pp. 1–40, 2016.

[235] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Cross-modal deep learning," *Proceedings of the 31st International Conference on Machine Learning*, pp. 1285–1294, 2014.

[236] J. V. Haxby, J. S. Guntupalli, A. C. Connolly, Y. O. Halchenko, B. R. Conroy, M. I. Gobbini, M. Hanke, and P. J. Ramadge, "A common, high-dimensional model of the representational space in human ventral temporal cortex," *Neuron*, vol. 72, no. 2, pp. 404–416, 2011.

[237] E. Schubert, "Continuous self-report methods," *Handbook of Music and Emotion: Theory, Research, Applications*, pp. 223–253, 2011.

[238] L.-C. Yang and A. Lerch, “Evaluation of music creativity and musical metacreation systems,” *Artificial Intelligence*, vol. 29, no. 1, pp. 100–117, 2018.

[239] A. Nijholt, D. Tan, G. Pfurtscheller, C. Brunner, J. d. R. Millán, B. Allison, B. Graimann, F. Popescu, B. Blankertz, and K.-R. Müller, “Brain-computer interfacing for intelligent systems,” *IEEE Intelligent Systems*, vol. 23, no. 3, pp. 72–79, 2008.

[240] E. R. Miranda, A. Brouse, and B. Boskamp, “Brain-computer music interfacing: a survey,” *Journal of New Music Research*, vol. 39, no. 1, pp. 1–21, 2010.