CLPS 1291 Final Project Report

Evan Li, Nick Masi, Theo McArn

§1. Abstract

The goal of our project was to apply the deep learning and computational neuroscience skills we have learned in this class and apply them to the modality of audio. We use data from Nakai et al.[1] who took fMRI brain scans of five subjects as they listened to songs from the gtzan dataset. Gtzan contains songs from 10 genres with 100 songs for each genre.[2] We then reimplement the process described by Denk et al. in their paper *Brain2Music*[3] where they take these fMRI brain scans and reconstruct the song that the subject was listening to while the neural data was recorded. We provide some samples of original recordings and reconstructions that we created.

§2. Methods

The core model used by *Brain2Music* is MusicLM,[4] a model also developed by Agostinelli and Denk et al., which inputs a text prompt and outputs a generated song based on the semantic meaning of the text. In order to create *Brain2Music* it was necessary to first train MusicLM. An open source implementation[5] of MusicLM was created by Phil Wang and this was adapted to be used for our project. Wang's implementation includes the necessary architecture but lacks any pretrained models. As a result, we attempted to train this large model using the Music Caption Dataset for *Brain2Music*.[6] This dataset is composed of 540 song snippets each which has a corresponding text description. These songs line up with the same songs listened to by the participants in the fMRI machine. These song-text pairs were necessary to train MusicLM. MusicLM is comprised of two components. The first component is the autoencoder MuLan, which is composed of an audio and text transformer. During

---

[1] Tomoya Nakai, Naoko Koide-Majima, and Shinji Nishimoto, Music Genre fMRI Dataset, 2021, OpenNeuro, [Dataset] doi: 10.18112/openneuro.ds003720.v1.0.0, https://openneuro.org/datasets/ds003720/.

[2] George Tzanetakis and Perry Cook, "Musical Genre Classification of Audio Signals," IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING 10, no. 5 (July 2002): 293–302, https://www.cs.cmu.edu/~gtzan/work/pubs/tsap02gtzan.pdf.

[3] Timo I. Denk, Yu Takagi, Takuya Matsuyama, Andrea Agostinelli, Tomoya Nakai, Christian Frank, and Shinji Nishimoto, "Brain2Music: Reconstructing Music from Human Brain Activity," 2023, arXiv, eprint 2307.11078, https://arxiv.org/pdf/2307.11078.pdf.

[4] Andrea Agostinelli, Timo I. Denk, Zalan Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank, "MusicLM: Generating music from text," 2023, arXiv, eprint 2301.11325, https://arxiv.org/abs/2301.11325.pdf

[5] Phil Wang, musiclm-pytorch, 2023, GitHub, [Source code], https://github.com/lucidrains/musiclm-pytorch

[6] NishimotoLab, Music Caption Brain2Music Dataset, 2023, Kaggle, [Dataset] https://www.kaggle.com/datasets/nishimotolab/music-caption-brain2music

training, the model takes in a song and a caption the two transformers attempt to create identical embeddings from these two input modalities. This way, during evaluation, only the input text prompt is embedded and propagated forward in the model to generate the new song. The second component is AudioLM, the predecessor to MusicLM, which is comprised of three transformer networks: Semantic, Coase, and Fine, as well as two pretrained auto-encoders: wav2vec, and Soundstream. In all this means that within MusicLM there were four submodules that all needed training. We had hoped to easily get MusicLM working in order to focus on its adaptation to *Brain2Music*. However, given the complexity and intricacies of MusicLM, we were forced to dedicate much more of our efforts to this portion of the project.

For AudioLM, the three transformers: Semantic, Coarse, and Fine were each trained for 1000 epochs and checkpoints were saved in order to load these models in for future use. However, for the fine transformer, we only used the checkpoint saved after 1 epoch because the loss quickly became nan after a few epochs. Mulan was also trained on multiple epochs but loss quickly plateaued. All of these models were trained using the Music Caption *Brain2Music* Dataset and the corresponding 540 songs in gtzan. This is a clear limitation as the original MusicLM was trained on around 50 million samples. We trained these networks using scripts[7] we wrote in Google Colab, in order to utilize the GPU compute offered.

Lastly, in order to replicate *Brain2Music*, the last step was to create the Linear Regression model which converts the fMRI scans into embeddings that can be used to pass into the music generating model(s). This was done by first passing all of the songs through MuLan in order to get the music embeddings. Then, each fMRI reading was paired with the correct Mulan song embedding of the song listened to during the scan. These pairing acted as the input-groundtruth pairs used to train our linear regression model, *W*, which was a single fully-connected linear layer with L2 regularization of weights. It was trained to reproduce the MuLan song embedding using only the fMRI data recorded when listening to that song. Since every brain is different, the model necessary to replicate the MuLan embeddings will be unique for every subject. As a result, we focused on training the linear regression for only one subject: Subject #1. We trained this network for 300 epochs. Then after training, We then apply *W* to novel fMRI data which was not used in the model's training. This generates music embeddings from the song which can then be passed into MusicLM in order to reconstruct the listened song. We trained on 50 songs and reconstructed 10. The generated 10 and the original songs they were trying to reconstruct are linked in the Results section.

[7] Evan Li, Nick Masi, Theo McArn, MusicLM.ipynb, 2023, Google Colab, [Source code] https://colab.research.google.com/drive/1PxH4eZvd0RiwKpdPI1YXdP1Uz8T9hnUT?usp=sharing

§3. Challenges

One of our first challenges was with preprocessing the fMRI data. Since none of us had previous experience working with fMRI data, we found it difficult to recreate the original preprocessing steps of *Brain2Music*. We ended up following the preprocessing steps from Kunkhe et al[8], which utilizes SPM12 to apply motion correction, distortion correction, slice timing correction, coregistration of functional and structural images, normalization, and smoothing. Additionally, we could not figure out in a timely manner on how to find regions of interest (ROI) in fMRI data. However, for future work, by isolating specific regions of interest within our fMRI data, this would greatly reduce the dimensionality of our input data and would remove extraneous features. This would reduce complexity and increase the performance of our model.

Another major issue was the fact that we often ran out of memory on Colab. This occurred when preprocessing or training on the high dimensional neural data and/or the large music files. This forced us to use limited amounts of data: less of gtzan than we could have and less of the neural data than was made available by Nakai et al. Additionally, and related to the previous issue, we suspect that the ROI's found and used in the neural data serve as hand-selected features which enable the authors to effectively utilize a simple linear regression for *W*. We attempted to compensate for the fact that we didn't use this by using a deeper MLP model with non-linearities for *W*, but even for a relatively simple 3-layer model with ReLU in between our Colab sessions would run out of memory and crash. As a result we used the same linear regression approach as Denk et al.

The *Brain2Music* paper was unclear on their implementation at many points, especially when discussing the linear regression they used to transform fMRI data into the MuLan music embedding space. All of this data was very high dimensional because of the 4D nature of fMRI data (3D + time) plus the additional dimensions for which test run and on what subject. This made the situation all the more confusing. What we have ultimately determined that the paper does is that for each song a subject listens to, which are 15s long, the fMRI data is recorded in 10 1.5s intervals. The songs however are split into 4 10s intervals which are then embedded with MuLan. To match the dimensions so that a single matrix transformation can go from fMRI data to song embeddings (learning this transformation is training the linear regression *W*), they downsample the fMRI data by averaging sets of seven (though the paper they say five for some reason) 1.5s intervals to get 10.5s intervals. There are 4 such continuous sets in each 15s interval, and these 4 averaged sets of fMRI data are matched to the 4 embeddings per song for the purpose of training *W*. This was unclear to us at first, so we use the same window size of 1.5s for song embeddings as we do for fMRI data, so

[8] Philipp Kuhnke, Markus Kiefer, Gesa Hartwigsen, Task-Dependent Recruitment of Modality-Specific and Multimodal Regions during Conceptual Processing, Cerebral Cortex, Volume 30, Issue 7, July 2020, Pages 3938–3959, https://doi.org/10.1093/cercor/bhaa010

we directly correspond unaveraged fMRI scan windows to the same time window in songs. This yields 10 windows per song (this is variable *r* as defined in Table 1 of §3.3 of the paper), rather than 4. Each window is mapped to a 128-dimensional embedding by our trained *W*, and the average of these 10 embeddings are what is passed in to AudioLM to generate a song conditioned on the neural data.

This use of AudioLM is a solution to another issue we faced when attempting to reimplement the work of Denk et al. The MusicLM library we use from Wang does not allow for conditioning on music embeddings, only text. However, the AudioLM library does. Once we train the MuLan quantizer using MusicLM, we can pass it into AudioLM so that it behaves in essentially the same way as MusicLM. We thus create a workaround and do our audio generation/song reconstruction by passing in embeddings to AudioLM with the MuLan quantizer instead of MusicLM.

§4. Results

Below we provide a table of 10 songs listened to by subjects which we reconstructed from their fMRI data using our model.

Original Song	Blues	Classical	Country	Disco	Hiphop	Metal	Jazz	Pop	Reggae	Rock
Reconstructed Song	1	2	3	4	5	6	7	8	9	10

§5. Conclusion

Our original research idea relied on the fact that we thought with *Brain2Music* that the MusicLM model was learning from the fMRI data. If this were the case, then the final transformer might potentially have some more humanistic understanding of songs, similar to Harmonization[9] for vision. However, it turns out that the linear regression they use to transform the fMRI data into the song embedding space (which is then passed into MusicLM) is the only place humanistic vision might be gained, as the MusicLM model is not trained with any neural data and is then frozen to use for inference. The linear regression uses the MuLan embeddings for songs as the ground truth, so MusicLM is given something in the shape of MuLan embeddings it is familiar with. This allows novel neural data at inference time to be mapped to the MuLan latent space, and then passes this into MusicLM (which has been trained in its standard manner with no neural data) so that a song can be generated. This means the novel addition of

[9] Thomas Fel, Ivan Felipe, Drew Linsley, and Thomas Serre, “Harmonizing the object recognition strategies of deep neural networks with humans,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, https://github.com/serre-lab/Harmonization.

Brain2Music was the simple linear regression model which learns a mapping from neural data to the MuLan embedding space, not anything to do with MusicLM! Unfortunately, it also meant there is likely no encoded human knowledge of music in MusicLM, though the MuLan embeddings might have some new human flavor to them. If that is the case, it would imply that the humanistic understanding of audio does exist in the linear regression model *W*.

One piece of future research we are very interested in is whether a linear probe can be added to *W* in order to perform music genre recognition (MGR) on the inputted song. This would investigate whether the model learning from fMRI data on listening to songs improves the model's performance on downstream audio tasks. Furthermore, because training of our *W* was on such limited song and neural data, it would be interesting to see the impact of greater scale (more data and more advanced models, such as a transformer instead of linear regression) on MGR or other downstream audio tasks.

Overall, we were not able to reconstruct songs at high fidelity nor accurately. This is due mostly to limited amounts of training data, and the lack of time and compute to even process and utilize all of the training data that we did have available. However, the three of us learned a tremendous amount about the state of the art of song generation, both at a theoretical level and with the practical implementation, as well as an innovative way to apply computational neuroscience techniques to the deep learning domain. We also developed our skills in the whole processing pipeline—from preprocessing to use in training models—of neural data which was a different modality (fMRI) than anything covered in class. Furthermore, we are very excited that we were able to generate songs at all, as this was a novel form of generative model for all of us.