Generating Mood Music

ADSP 31018 | Bose | Spring 2025
Group Deep Fried Neurons | Lauren Adolphe, Aneesha Dasari, Jazil Kalim, Pat Ohea, Maxine Wu
Source code: https://github.com/jazilkalim/Generating-Mood-Music

THE UNIVERSITY OF CHICAGO

Executive Summary

Objective

Build a generative deep learning model that can create short music clips conditioned on emotional mood input.

For simplicity, we focus on four emotions:

- sad
- happy
- relaxed
- angry

Approach

Used DEAM dataset with valence and arousal annotations.

Converted audio into mel-spectrograms for model training.

Explored different potential models, such as LSTM-GAN and autoregressive GAN.

Outcome

Finalized a conditional GAN model for audio generation.

Generated music clips which were metallic-sounding.

The outcome is achievable but computational resources and more exploration is required.

Problem Statement

Goal

Build a generative deep learning model that takes a mood input — such as happy, sad, angry, or calm — and outputs a short music clip that reflects that mood.

The Importance of Music

Music is a universal emotional language. It shapes our mood, focus, and memory. Music is increasingly being used in AI:

- therapy & mental health apps
- adaptive game soundtracks
- personalized playlists.

Motivations

Most generative music models ignore emotional intent. Generating audio with a mood in mind has real-world creative potential. Applications include:

- Dynamic film/game soundtracks that adapt to emotion
- Tools for composers, creators, or music therapists
- Advances generative AI by fusing emotion understanding with raw audio generation

The Data

Database for Emotional Analysis using Music (DEAM)

Content

★ 2,000+ excerpts and full-length songs annotated with valence and arousal values both continuously (per half-second) and aggregated over the whole song

Emotion Dimensions

★ **Arousal** – energy/intensity labeled per second
★ **Valence** – positivity/pleasantness labeled per second
★ **Mean Arousal** – energy/intensity for a track
★ **Mean Valence** – positivity/pleasantness for a track

Assumptions

- Discrete moods (e.g., happy, sad) can be mapped to static (valence, arousal) pairs
- Emotional tone can be inferred and synthesized from short (5–10 sec) audio clips
- Mel-spectrograms retain key musical features (e.g., rhythm, energy, timbre) needed for emotional interpretation
- Generative models can be conditioned effectively on low-dimensional mood inputs

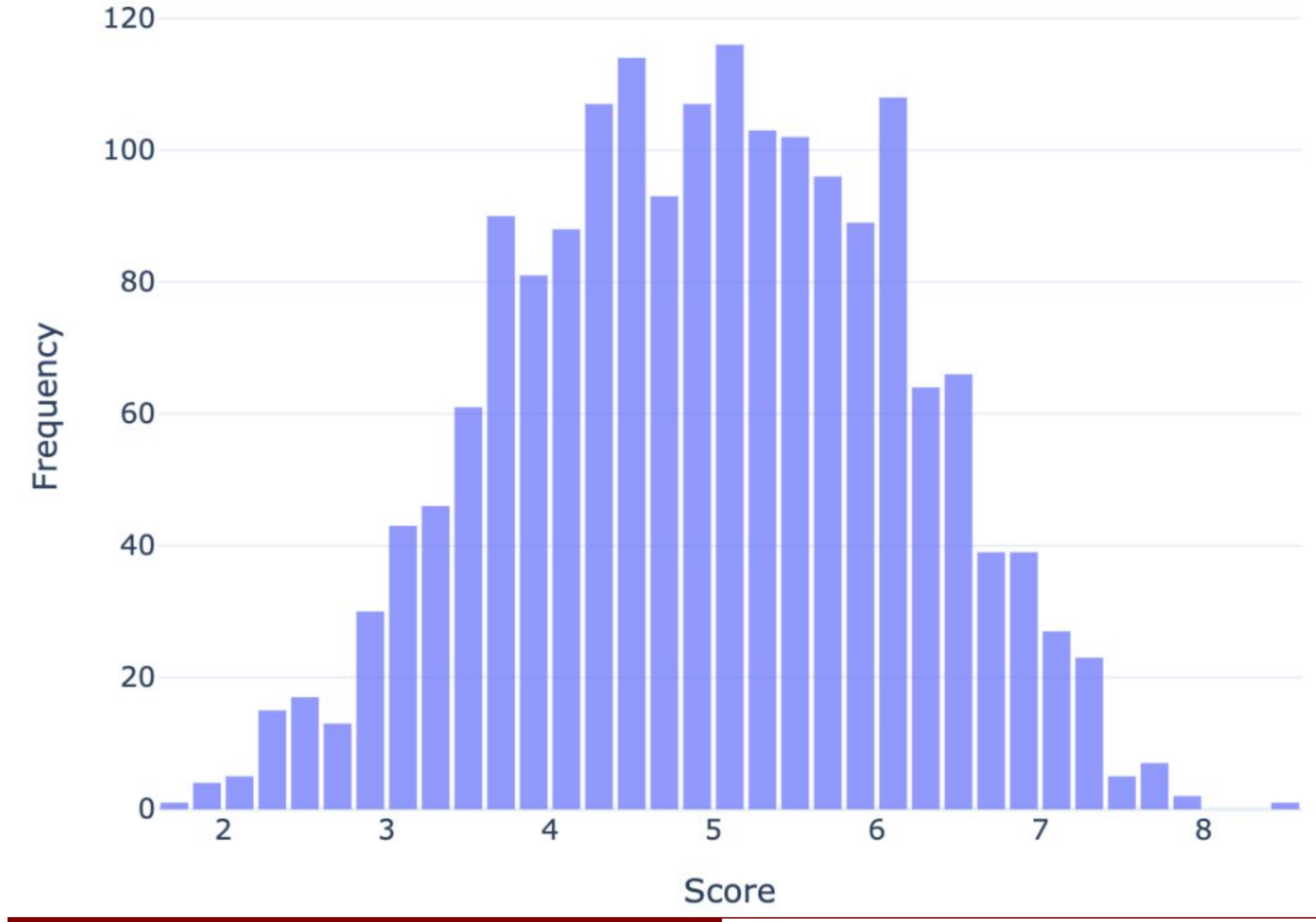
THE UNIVERSITY OF CHICAGO

Exploring the Data

There is enough variety in the valence and arousal of the songs to use these values to train the models.

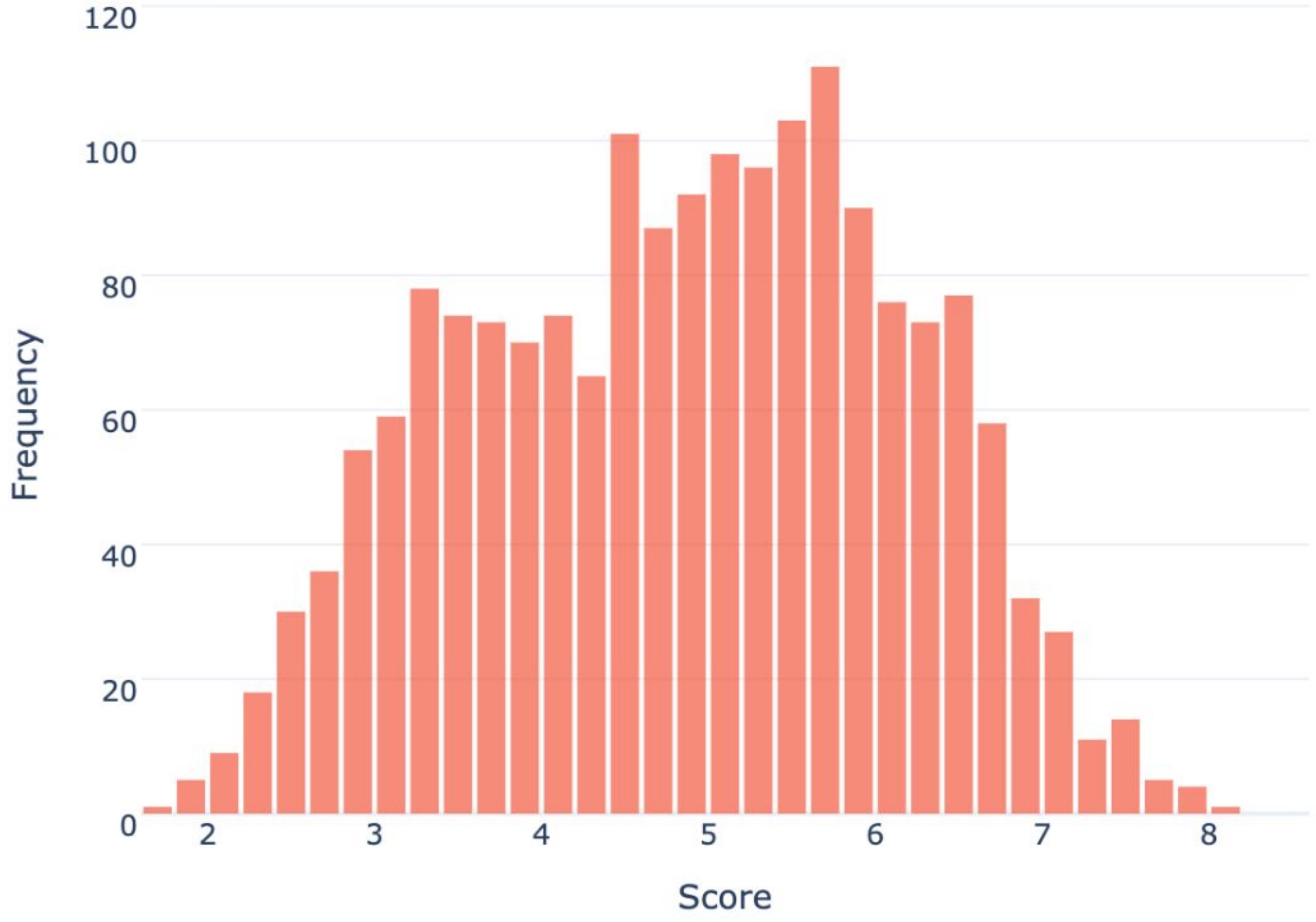
Example Audio	Mean Valence: 5.7, Mean Arousal: 5.5

Distribution of Mean Valence



minimum	1.6
maximum	8.4
average standard deviation	1.50

Distribution of Mean Arousal

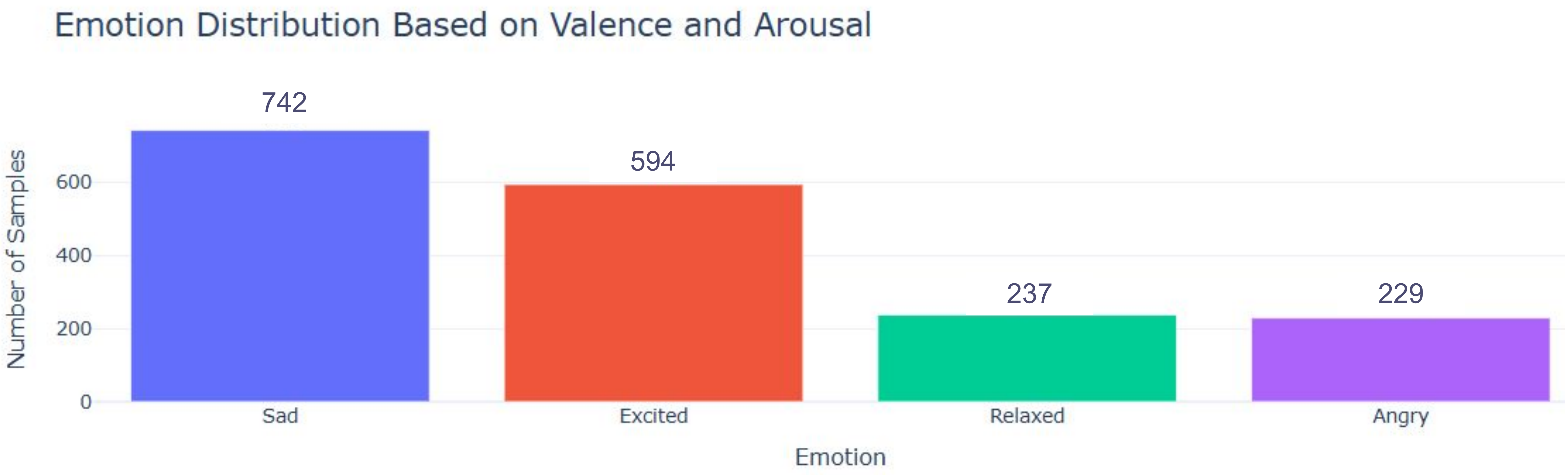


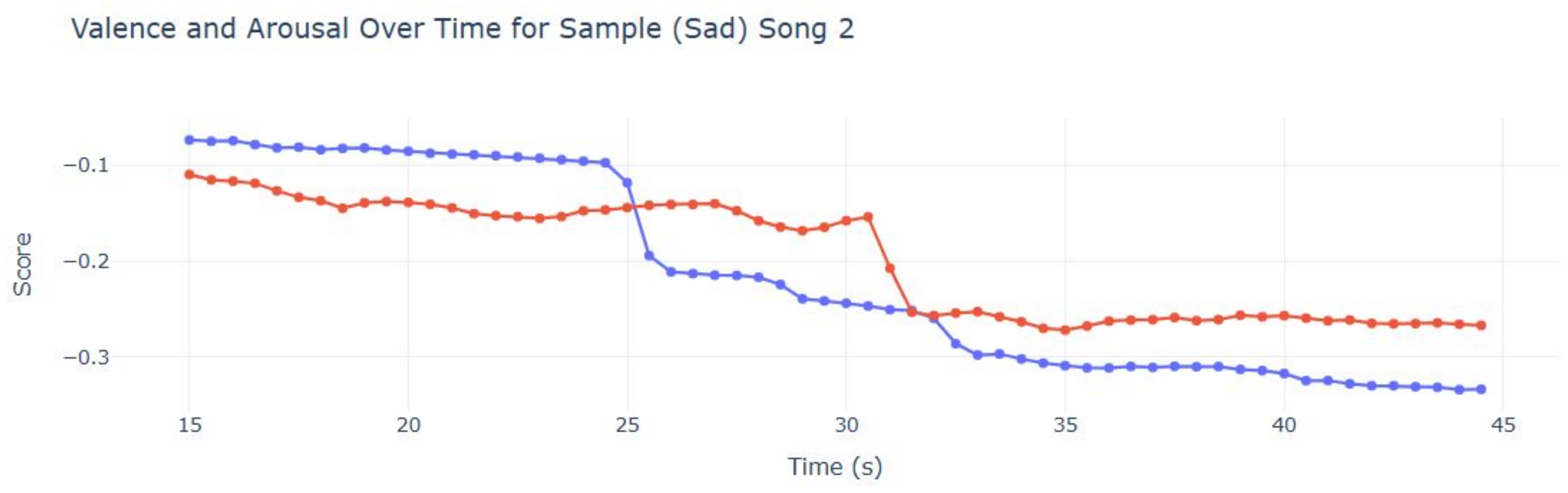
minimum	1.6
maximum	8.1
average standard deviation	1.46

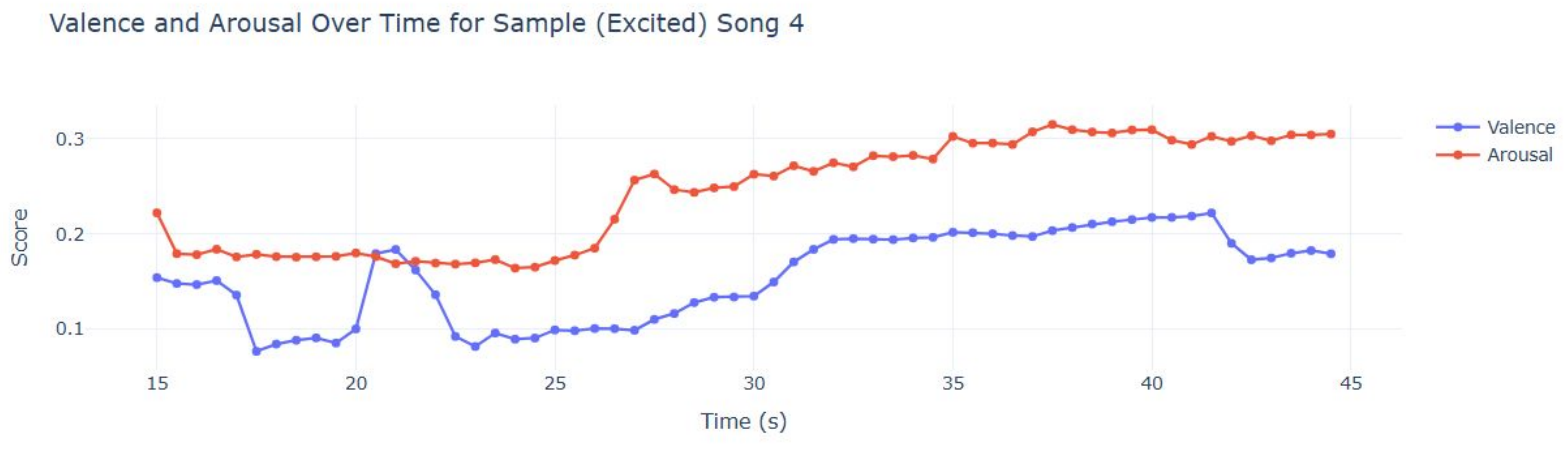
Exploring the Data

For this application, we are looking at identifying emotion through valence and arousal. For simplicity, we assess the **normalized valence and arousal values** (-1 to 1) and begin by focusing on **four main emotions**. This is used to further understand the data.

Emotion	Valence	Arousal
Sad	low	low
Excited	high	high
Relaxed	high	low
Angry	low	high







Feature Engineering

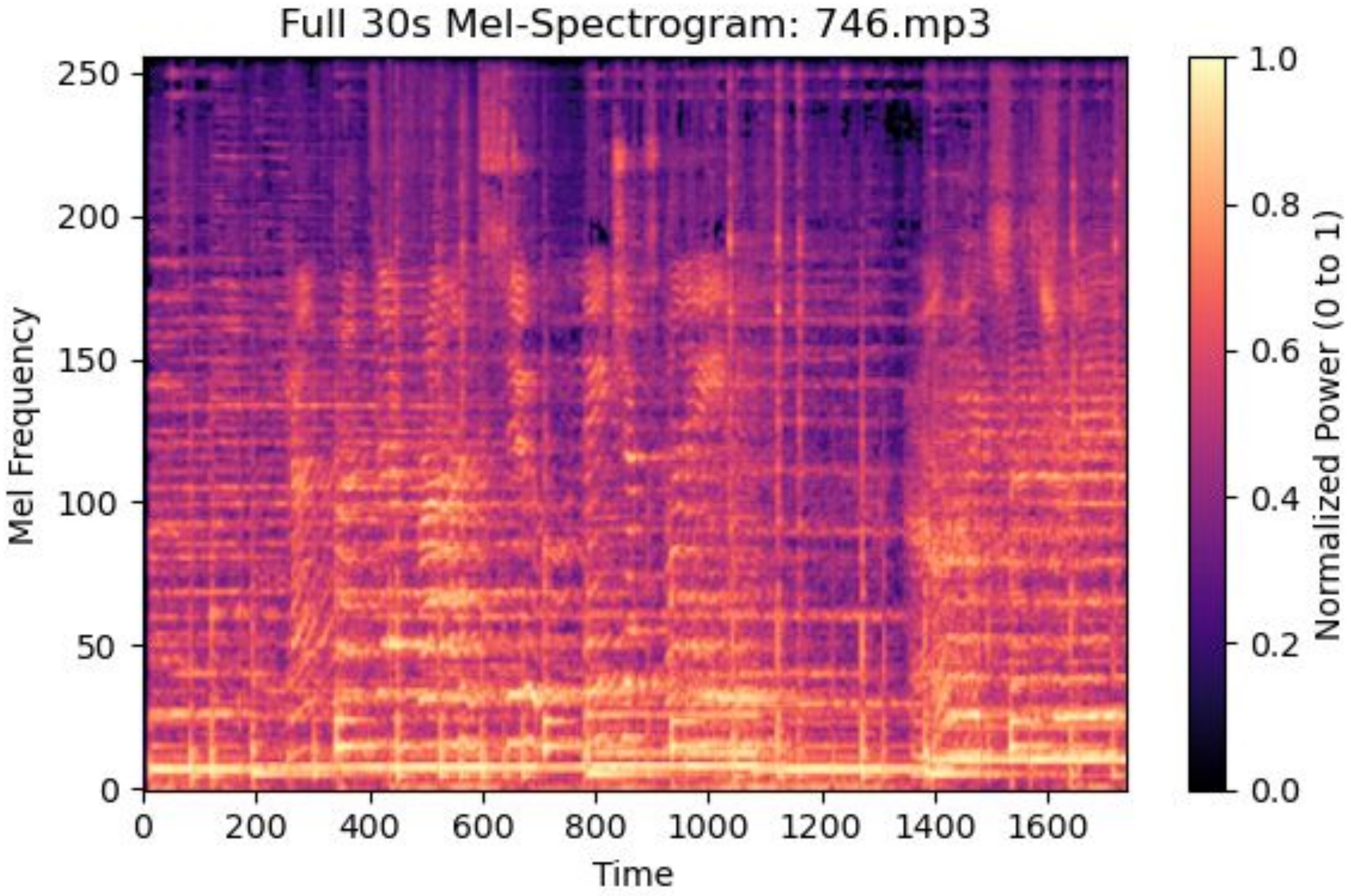
Primary feature engineering focused on normalizing data and creating mel-spectrograms from the audio.

From Audio to Mel-Spectrograms

Mel-spectrograms are a **2D time-frequency representation of sound**.

Why use mel-spectrograms?

- can preserve emotional cues and reflect human perception of pitch, tempo, and timbre
- easy to visualize and compatible with CNNs, LSTMs, and GANs
- already commonly used in music emotion recognition and audio generation



Explored Approaches

#	Type	Description	Decision
1	Conditional VAE	• Encodes input + condition to a latent space • Allows **smooth latent sampling**, but reconstructions were **blurry or noisy** • Difficult to maintain musical coherence	❌
2	LSTM-Based GAN	• Started with an **RNN-LSTM generator** • Replaced with an **autoregressive feedforward model** • Encountered **unstable training**, **mode collapse**, and **gradient saturation**	❌
3	Autoregressive GAN	• Generates each slice using previous output + condition • Reasonable **emotional alignment** • More stable, but still produces noisy output and **repetitive sequences**	❌
4	cGAN	• Generator uses previous **output + emotion input** • Discriminator checks if output matches (v,a) condition • Best output compared to other models	✔

★ All models, even our choice, remained underfit, as the output audios and spectrograms elucidate.

Chosen Solution

cGAN

Training

Reduce Variance			Reduce Bias
label smoothing real = 0.9	**gaussian noise** for robustness	**dropout regularization** to improve generalization	**four generator** training steps **per discriminator** training step

Choice Rationale

Conditionally Guided Generation
Directly incorporates mood input (valence & arousal), allowing targeted emotion control

Training Stability
More stable than standard GANs due to regularization techniques (e.g., label smoothing, noise injection)

Autoregressive Architecture
Generates spectrogram slices step-by-step, reducing sudden artifacts and improving temporal consistency

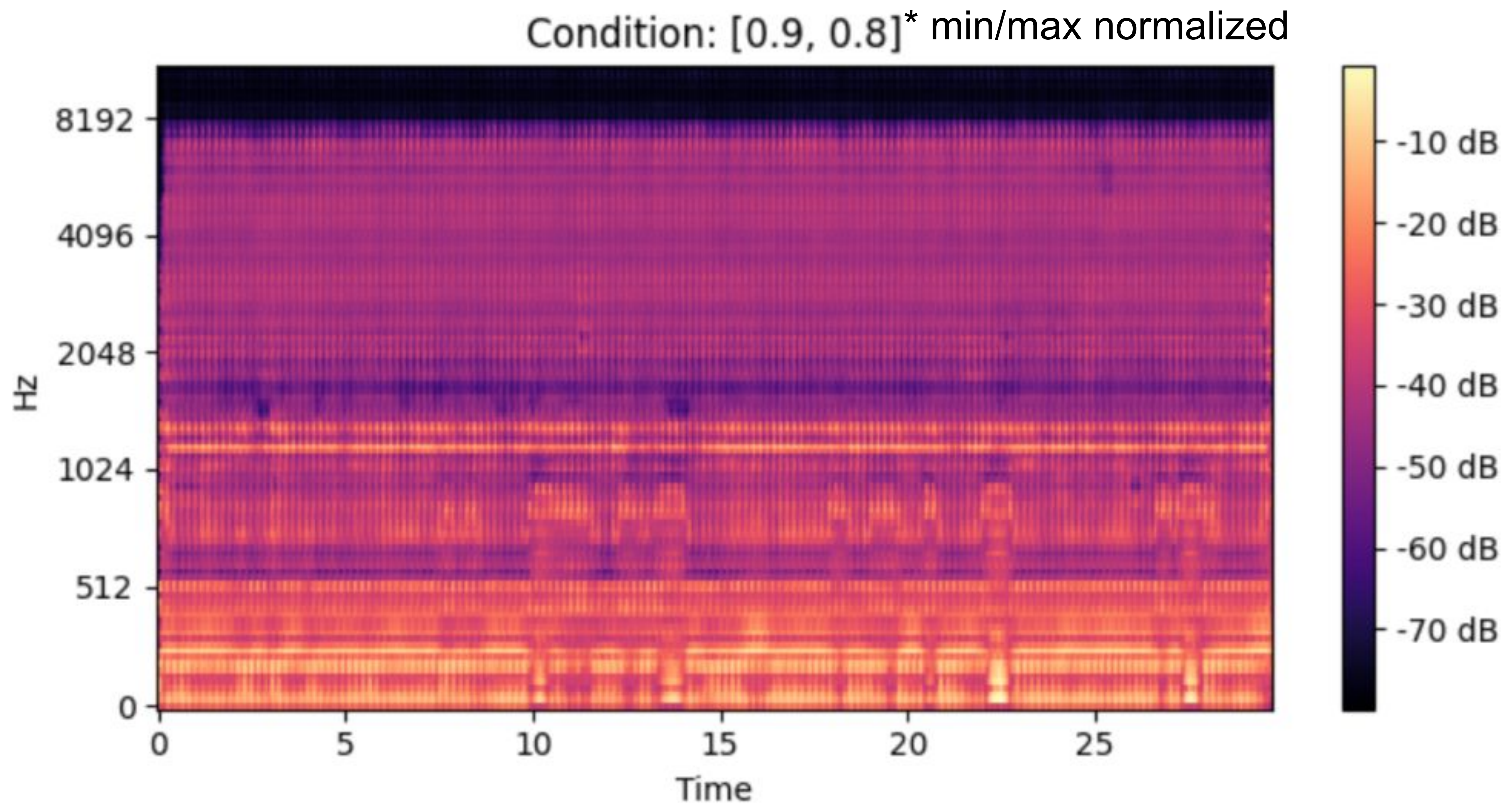
Better Sample Quality
Produced more coherent, emotionally aligned, and musically structured audio compared to CVAE or LSTM-GAN

Flexible for Inference
Allows generation of novel music for arbitrary mood conditions without needing real input examples

Results

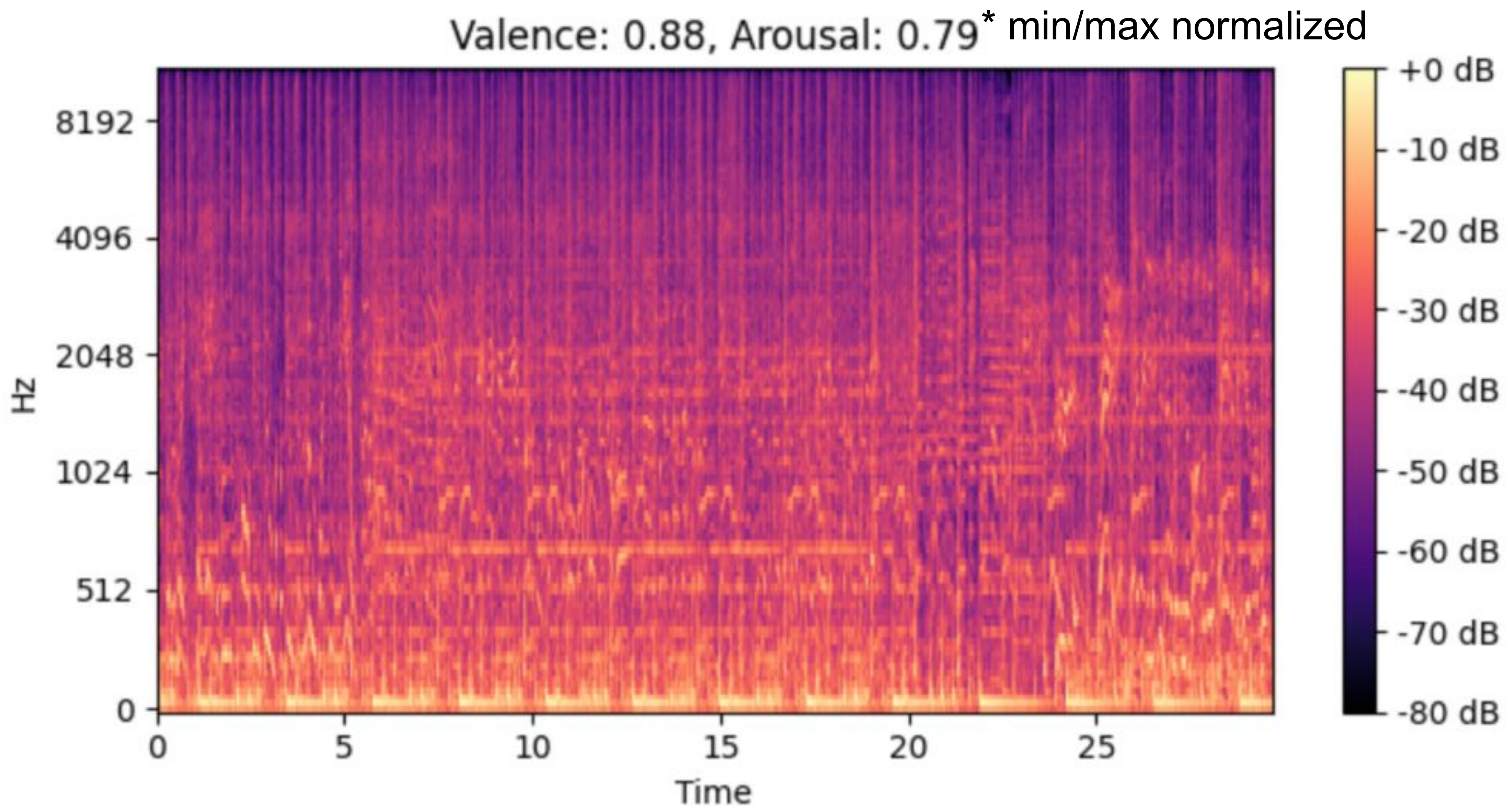
cGAN Generated

Mean Valence = 7.92
Mean Arousal = 7.11



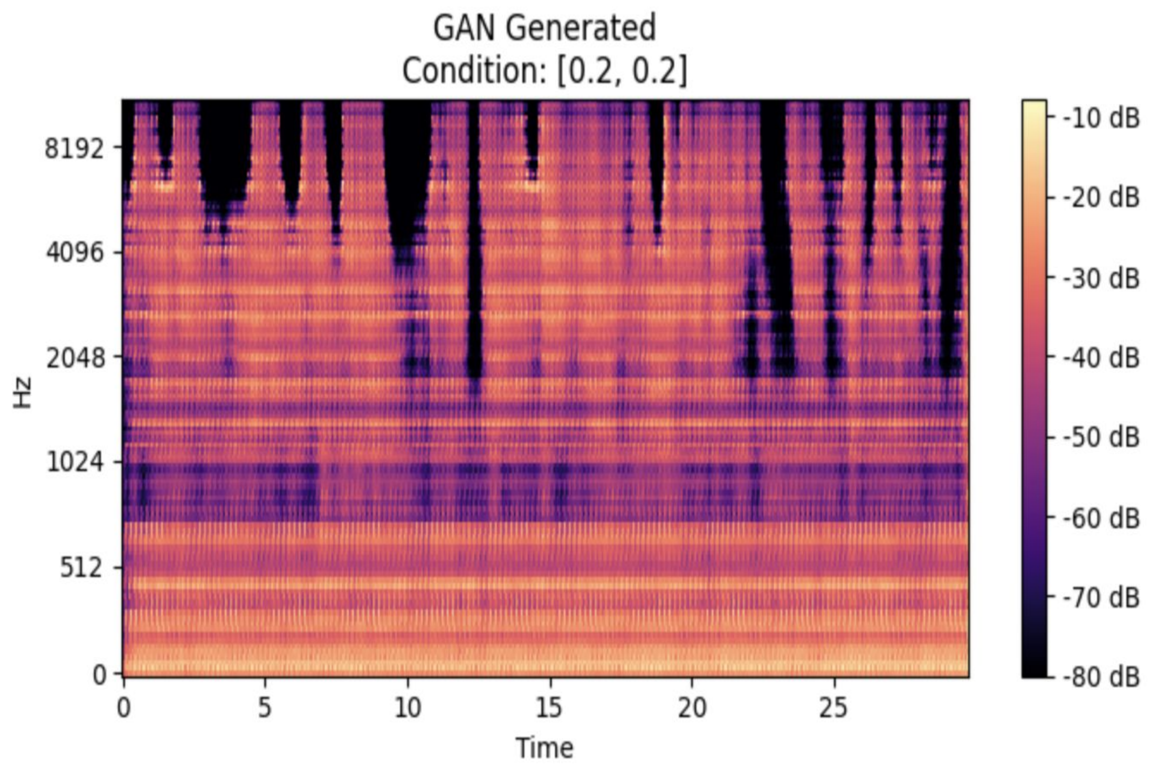
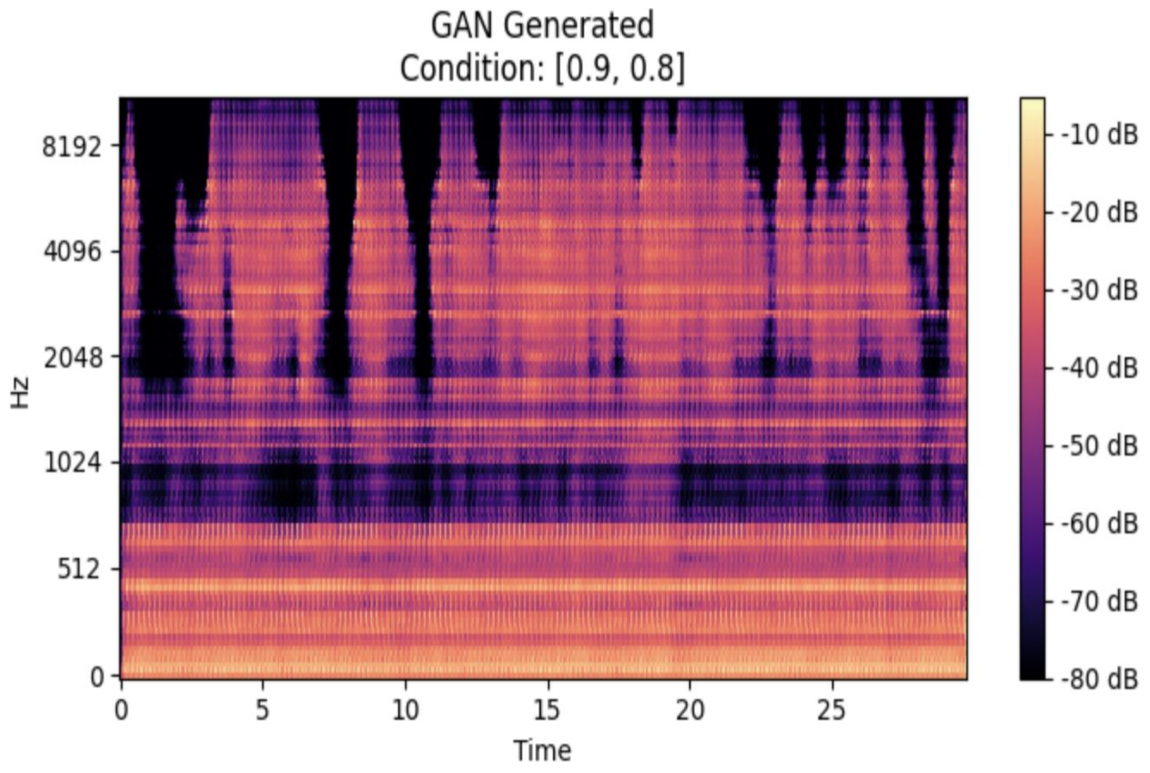
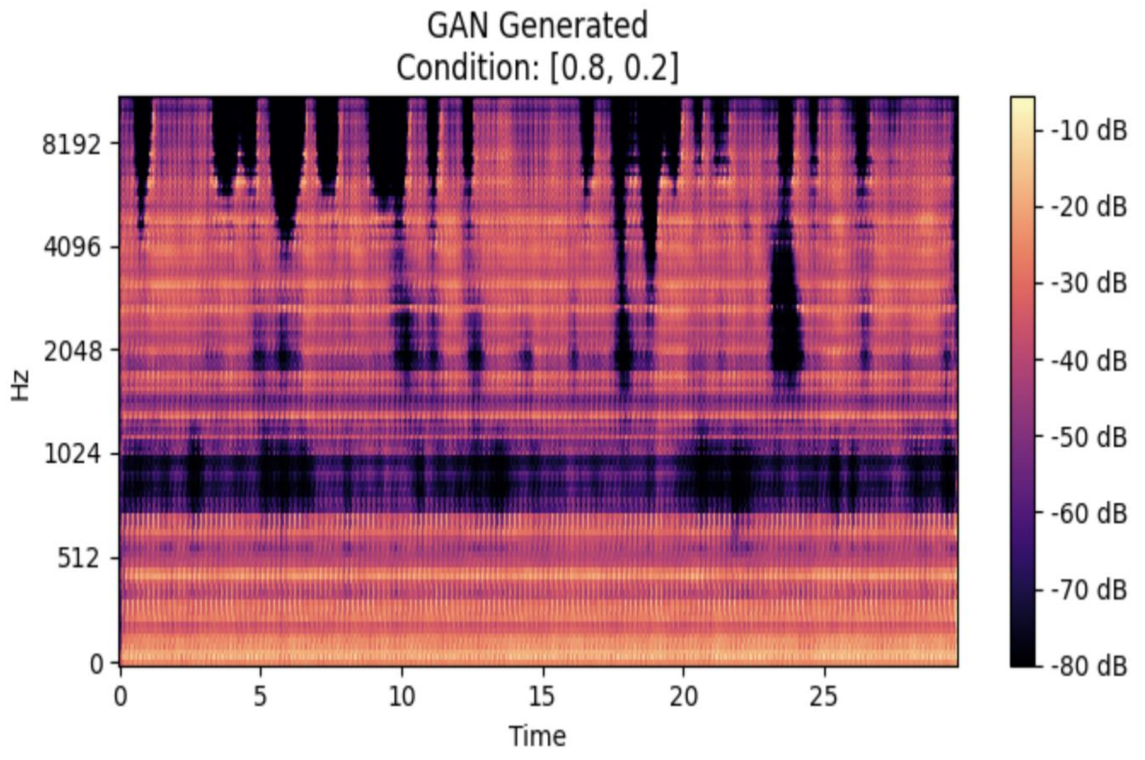
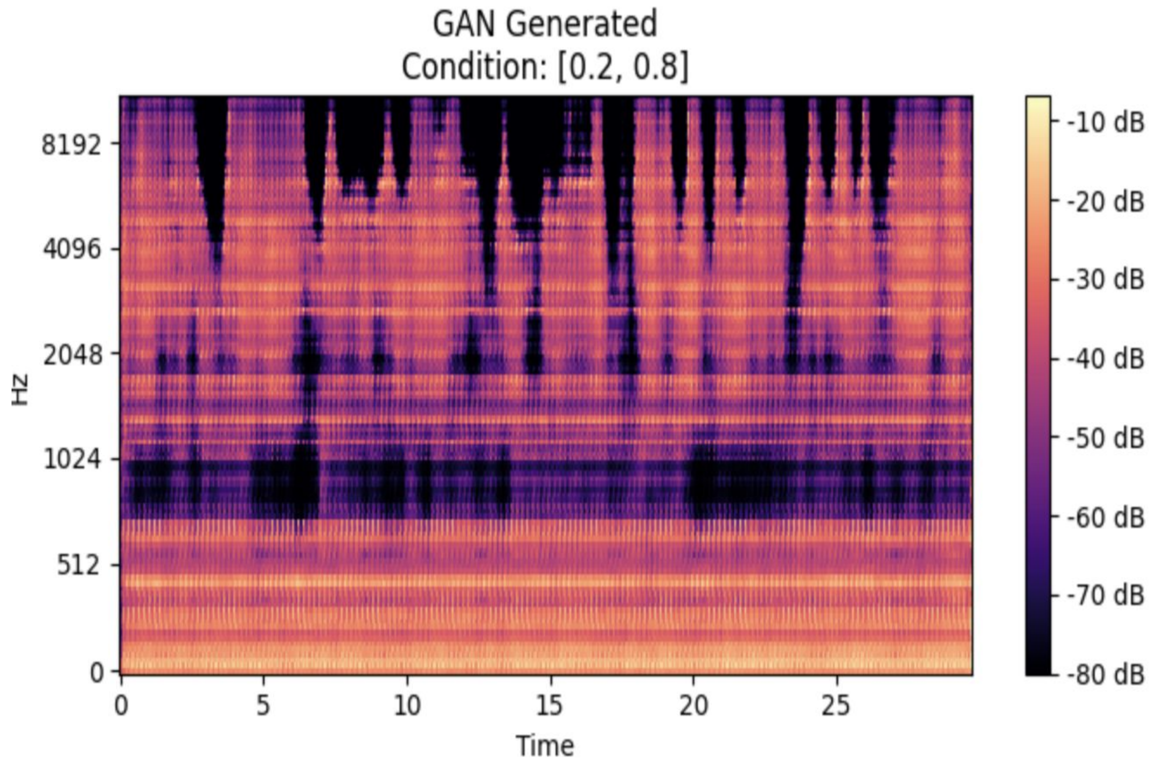
Closest Real Song

Mean Valence = 8.1
Mean Arousal = 7.11



Results

We compare generated clips for each emotion to evaluate the model subjectively.

Sad	Excited	Relaxed	Angry
Valence = low Arousal = low	Valence = high Arousal = high	Valence = high Arousal = low	Valence = low Arousal = high
			
MAE: 0.3599 MSE: 0.2096	MAE: 0.4514 MSE: 0.3050	MAE: 0.3734 MSE: 0.2072	MAE: 0.3355 MSE: 0.1797

Interlude: A more pleasant AI Tune

As a palette cleanser, here's some AI-generated music that won't hurt your ears.

Challenges

Temporal Coherence

Frame-wise generation led to abrupt, disjointed transitions.
→ Introduced **autoregressive generation** using previous mel slice and emotion input for smoother continuity.

Label Noise

Valence/arousal annotations were subjective and variable.
→ Used DEAM's per-second labels with **temporal smoothing**, but emotional ground truth remains inherently ambiguous.

GAN Instability

Faced **mode collapse** and discriminator overpowering the generator.
→ Mitigated with label smoothing, dropout, Gaussian noise, and introducing **multiple G steps per D update**.

Compute Limitations

Model with **350M+ parameters** strained memory and iteration speed.
→ Even on an **A100 GPU**, required careful **batch sizing** and **gradient tuning** for stability.

Future Work

Architectural & Tuning Improvements

Use a neural vocoder like **HiFi-GAN**, upgrade to a U-Net generator with deeper conditioning, and **add perceptual audio losses for realistic, structured output**.

Use Pre-trained Audio Models

Leverage pretrained models like MusicLM, Jukebox, or AudioCraft to **boost fidelity, and enable advanced controls** like text-to-music and style transfer.

Full Song Generation

Extend generation to **full-length audio clips with evolving emotional profiles**, using temporal dynamics or hierarchical modeling to preserve musical coherence and structure over time.

Realtime Emotion Input

Integrate **biosignal or facial emotion detection** to enable adaptive music in games, therapy, and **immersive experiences**.

Conclusion

Mel-spectrograms combined with valence/arousal conditioning enable meaningful emotion-aware music generation.

The final cGAN model showed the best trade-off between training stability and expressive control.

Pleasant and “real” music generation is much more complicated than what can be accomplished for a class project!

AI music generation has the potential to make impacts in the arts and healthcare.

Thank You

THE UNIVERSITY OF
CHICAGO

Appendices

THE UNIVERSITY OF CHICAGO

1. Resources & References

Project Code on Github: https://github.com/jazilkalim/Generating-Mood-Music

DEAM resources:

Kaggle dataset:
https://www.kaggle.com/datasets/imsparsh/deam-mediaeval-dataset-emotional-analysis-in-music

Original DEAM database: https://cvml.unige.ch/databases/DEAM/

Python packages used:
audioread, glob, kagglehub, librosa, numpy, os, pandas, pickle, plotly, pydub, random, tensorflow

Disclosure:
This analysis was conducted with assistance from ChatGPT (OpenAI, 2025) for code generation and troubleshooting.

2. cGAN Architecture

Discriminator

Layer (type)	Output Shape	Param #	Connected to
input_layer_2 (InputLayer)	(None, 128, 2560, 1)	0	-
conv2d (Conv2D)	(None, 64, 1280, 64)	1,088	input_layer_2[0]…
leaky_re_lu (LeakyReLU)	(None, 64, 1280, 64)	0	conv2d[0][0]
dropout_4 (Dropout)	(None, 64, 1280, 64)	0	leaky_re_lu[0][0]
conv2d_1 (Conv2D)	(None, 32, 640, 128)	131,200	dropout_4[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 32, 640, 128)	0	conv2d_1[0][0]
dropout_5 (Dropout)	(None, 32, 640, 128)	0	leaky_re_lu_1[0]…
flatten (Flatten)	(None, 2621440)	0	dropout_5[0][0]
input_layer_3 (InputLayer)	(None, 2)	0	-
concatenate_1 (Concatenate)	(None, 2621442)	0	flatten[0][0], input_layer_3[0]…
dense_1 (Dense)	(None, 1)	2,621,443	concatenate_1[0]…

Total params: 2,753,731 (10.50 MB)
Trainable params: 2,753,731 (10.50 MB)
Non-trainable params: 0 (0.00 B)

Generator

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 256)	0	-
input_layer_1 (InputLayer)	(None, 2)	0	-
concatenate (Concatenate)	(None, 258)	0	input_layer[0][0… input_layer_1[0]…
dense (Dense)	(None, 1310720)	339,476,4…	concatenate[0][0]
reshape (Reshape)	(None, 8, 160, 1024)	0	dense[0][0]
conv2d_transpose (Conv2DTranspose)	(None, 16, 320, 512)	8,389,120	reshape[0][0]
batch_normalization (BatchNormalizatio…	(None, 16, 320, 512)	2,048	conv2d_transpose…
dropout (Dropout)	(None, 16, 320, 512)	0	batch_normalizat…
re_lu (ReLU)	(None, 16, 320, 512)	0	dropout[0][0]
conv2d_transpose_1 (Conv2DTranspose)	(None, 32, 640, 256)	2,097,408	re_lu[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 32, 640, 256)	1,024	conv2d_transpose…
dropout_1 (Dropout)	(None, 32, 640, 256)	0	batch_normalizat…
re_lu_1 (ReLU)	(None, 32, 640, 256)	0	dropout_1[0][0]
conv2d_transpose_2 (Conv2DTranspose)	(None, 64, 1280, 128)	524,416	re_lu_1[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 64, 1280, 128)	512	conv2d_transpose…
dropout_2 (Dropout)	(None, 64, 1280, 128)	0	batch_normalizat…
re_lu_2 (ReLU)	(None, 64, 1280, 128)	0	dropout_2[0][0]
conv2d_transpose_3 (Conv2DTranspose)	(None, 128, 2560, 64)	131,136	re_lu_2[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 128, 2560, 64)	256	conv2d_transpose…
dropout_3 (Dropout)	(None, 128, 2560, 64)	0	batch_normalizat…
re_lu_3 (ReLU)	(None, 128, 2560, 64)	0	dropout_3[0][0]
conv2d_transpose_4 (Conv2DTranspose)	(None, 128, 2560, 1)	577	re_lu_3[0][0]

Total params: 350,622,977 (1.31 GB)
Trainable params: 350,621,057 (1.31 GB)
Non-trainable params: 1,920 (7.50 KB)