**Melodic Machines:**

[**Github Link**](https://github.com/ZahFay/CSCI1470-Melodic-Machines)

**Title**: Melodic Machines

**Who**:

Qien Lin (qlin23)

Alzahra Fayie (afayie)

Liang Man-Fang (mliang23)

Zetao Wu (zwu119)

**Introduction**: What problem are you trying to solve and why?

* Music creation is both an art and a science: crafting lyrics that feel “authentic” to a particular artist and producing audio that captures their signature sound can require extensive domain expertise and manual effort. Recent advances in deep generative models—transformers for text and diffusion models for images—open the door to automating these creative tasks.
* In this project, Melodic Machines, we explore a two‑stage framework for artist‑conditioned music generation. First, a transformer‑based lyric generator takes as input an artist’s name and produces novel lyrics in their stylistic voice. Second, a stable‑diffusion model operates in the spectrogram domain to synthesize new audio textures that mimic the artist’s sonic palette. By coupling these modules, our system can output both the words and the sound of a “new” song in a target artist’s style—streamlining creative workflows and offering insights into the interplay between lyrical content and audio style.

**Related Work**: Are you aware of any, or is there any prior work that you drew on to do your project?

* When we first came up with the idea, we actually were unaware of whether or not it had been implemented before. As a whole, we couldn’t find if our architecture of the two-model project has been done before, but we have found some implementations of each model separately with some variations.
* We found a website called Riffusion that claims to use spectrogram generation with stable diffusion, but we have no access to its source code. However, it takes in music descriptions/genres/etc. and produces full-fledged songs with clear lyrics instead of our model which takes in an artist as input and produces an audio sample in their “style”. Riffusion can be found [here](https://www.riffusion.com/library/my-songs) and a brief explanation of its architecture can be found [here](https://flowingdata.com/2022/12/16/stable-diffusion-spectrogram/).
* Lyric generation using transformers doesn’t seem like a completely novel idea either as we have found implementations of it online but they use GPT2 pretrained models. A medium article showing lyric generation with GPT2 can be found [here](https://arturorey.medium.com/artist-based-lyrics-generator-using-machine-learning-eb70dc4fb993). A hugging face fine-tuning of GPT2 for lyric generation can be found [here](https://huggingface.co/SpartanCinder/GPT2-finetuned-lyric-generation).

**Data**: What data are you using (if any)?

* Since we were not re-implementing a paper, we had to search for appropriate datasets ourselves for both generation tasks. However, since a dataset with the exact data we need to train our models on didn’t exist, we had to do significant preprocessing to have our data ready for training.
* Our initial plan was to train both our models on the common artists between the audio dataset FMA and lyric dataset MSD. We ran into issues during preprocessing the data for that as the artist intersection turned out to be very small (about 900 artists) and the lyrics stored in the MSD dataset were in bag-of-words format so it was not appropriate to be used as training data for our transformer model, where word order is integral to its design and the production of comprehensible lyrics.
* To resolve those issues, we switched to web-scraping the ground-truth lyrics of FMA dataset songs from the Genius API and downloaded famous artist lyrics from a kaggle dataset. The 30 second sample for famous artists will possibly be web-scraped using the Spotify API.
* Datasets/APIs used: [Famous artists lyrics](https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset?resource=download), [Free Music Archive (FMA)](https://github.com/mdeff/fma), [MillionSongDataset (MSD)](http://millionsongdataset.com/), [Genius API](https://docs.genius.com/), [Spotify API](https://developer.spotify.com/documentation/web-api)

**Methodology**: What is the architecture of your model?

* Lyric generation transformer model:
  + We plan on using a decoder-only transformer that is similar to a GPT-style model to generate lyrics conditioned on the artist identity:
    - Input: Artist name preprended as a special token to lyric sequences
    - Tokenizer: Subword-based tokenizer
    - Architecture:
      * Embedding layer
        + Token + positional + artist context embedding
      * 4-6 Transformer decoder blocks
        + Multi-head self-attention
        + Feed-forward network
        + LayerNorm + Dropout
      * Output linear layer projecting to vocab size
    - Training:
      * Loss: Cross Entropy Loss with teacher forcing (similar to RNN)
      * Optimizer: AdamW
      * Sampling Top-k sampling
* Spectrogram generation stable diffusion model:
  + We can implement a conditional stable diffusion model to generate spectrograms representative of an artist’s style. Instead of generating raw waveforms, which might be hard to model due to high dimensionality, we can operate with the mel spectrogram domain, which provides a more compact and perceptually relevant representation of the audio:
  + Input:
    - Text Conditioning: Artist names are tokenized and embedded using a pre trained text encoder (CLIP or custom Transformer/GRU)
    - Condition Injection: Text embeddings are fed into the U-Net via cross-attention layers or feature concatenation
  + Target:
    - 2D Mel Spectrograms: For each song in the dataset, we extract a log-scaled mel spectrogram (with another library or 3rd party tool e.g. librosa)
    - Each spectrogram can represent a 30-sec clip as a matrix of shape that is padded to a fixed dimension
  + Architecture:
    - We can implement a U-Net-based denoising architecture:
    - Encoder: Several downsampling blocks with convolution, attention layers, and residual connections
    - Bottleneck: Self-attention and cross-attention layers for conditioning on the artist embedding
    - Decoder: Upsampling blocks mirroring the encoder; skip connection to preserve local structure
  + Noise can be added at various timesteps, and the model is trained to iteratively denoise the spectrogram from pure Gaussian noise using DDPM
  + Conditional inputs are injected:
    - We can do this through either cross-attention at multiple layers or concatenation with U-Net features
  + Training:
    - Loss: Mean squared error between predicted and ground-truth noise at each denoising step
    - Noise scheduling: Linear or cosine beta schedule for optimal convergence
    - Dataset: FMA tracks grouped by artist to capture style-specific characteristics
  + Generation / Inference:
    - Sample noise: tensor with random Gaussian noise of shape
    - Condition: Provide the artist's name as a prompt to the text encoder and get the embedding
    - Denoise: Run the diffusion model over timestamps; progressively transform noise into a coherent mel spectrogram
    - Convert to audio:
      * Use some sort of Python Library (e.g., librosa) to invert the mel spectrogram into a time-domain waveform
      * Or we can use a pre-trained neural vocoder

**Metrics**: What constitutes “success?”

* What experiments do you plan to run?
  + Varying model prompts and evaluating whether generated outputs differ in stylistic ways
    - Inputting different artist names
  + Perform ablation studies by training on different components or models
    - Only FMA artists
    - FMA + famous artists
  + For lyrics, we can test temperature and top-k sampling strategies to balance creativity vs coherence
* For most of our assignments, we have looked at the accuracy of the model. Does the notion of “accuracy” apply for your project, or is some other metric more appropriate?
  + The notion of traditional accuracy might not necessarily be meaningful here as we are not solving a classification or prediction task with a fixed ground truth. Instead, we are focusing on generative creative outputs that are stylistically aligned with a given artist
  + Thus, we want to rely on qualitative metrics like human evaluation of fluency and style consistency. Success in our project is not being accurate in a traditional sense, but rather about generating convincing, artist-aligned creative content that resonates with human perception and personal opinions.
* If you are implementing an existing project, detail what the authors of that paper were hoping to find and how they quantified the results of their model.
  + I believe that we are not implementing an existing project
* If you are doing something new, explain how you will assess your model’s performance.
  + Since our project attempts to combine to generative models, transformer for artist conditioned lyric generation and stable diffusion model, for spectrogram-based audio synthesis, we are mainly relying on qualitative evaluation to assess performance
  + For the lyric generation model, we will evaluate outputs based on:
    - Stylistic consistency with target artist
    - Fluency and coherence of the generated text
    - Creativity and variability across multiple samples
  + For the spectrogram generation model, we will
    - Convert the generated spectrograms back to audio (maybe inverse transformations)
    - Conduct informal listening tests to judge whether the audio aligns with the style of input artist
    - Compare visual patterns between generated and real spectrograms
  + While we may use proxy metrics like perplexity for the lyric model, and inspect spectrogram similarity visually, the primary assessment method is more of a subjective human evaluation, focusing on how authentic, artist-aligned and perceptually convincing the outputs are
  + A main concern we have with our stable diffusion model is the long training time, so fine-tuning might not be possible.
* Base goal: complete our stable diffusion model along with a few spectrograms to audio conversions so they can be subjectively evaluated. Only FMA data is used.
* Target goal: complete both models with the transformer having acceptable outputs along with a few spectrogram conversions and lyric outputs to be subjectively evaluated. Only FMA data is used.
* Stretch goal: complete both models with fine-tuning to ensure “good” outputs along with a considerable amount of spectrogram conversions and lyric outputs to be subjectively evaluated. Both FMA data and famous artist data is used.

**Ethics**: Choose 2 of the following bullet points to discuss; not all questions will be relevant to all projects so try to pick questions where there’s interesting engagement with your project. (Remember that there’s not necessarily an ethical/unethical binary; rather, we want to encourage you to think critically about your problem setup.)

* What broader societal issues are relevant to your chosen problem space?

Since our project intersects with debates about AI-generated art replacing human creativity. By mimicking the artists’ styles, the model could inadvertently devalue original artistic labor or blur lines between inspiration and imitation. Also, we found that if trained on artists from marginalized communities without context, the model might reduce culturally significant styles to mere inputs, which might raise ethical concerns about ownership.

* Who are the major “stakeholders” in this problem, and what are the consequences of mistakes made by your algorithm?

Artists whose styles are being replicated: Original musicians whose unique creative expressions are being mimicked by the AI system are the primary stakeholders.

* + Misrepresentation of their artistic style
  + Potential reputational damage if generated content is offensive or low-quality
  + Possible loss of income or creative control

Music consumers/listeners: People who engage with AI-generated music.

* Being misled about the authorship of creative works
* Forming incorrect impressions about artists' styles or messages

Music industry professionals: Including record labels, producers, and promoters.

* Market disruption if AI-generated content floods distribution channels
* Potential shifts in how music creation is valued financially
* Since our project works with creative material, copyright concerns were our top priority. In terms of copyright concerns, we tried to avoid infringing on artists and their creative works by using ethical datasets such as FMA and training our models on 30 second samples instead of their full discography. However, that greatly reduced the relatability of our project since most famous artists were not participants in the FMA datasets.
* Since we had to resort to web-scraping after running into issues with the MSD dataset, we had a few concerns on how to ethically source ground-truth lyrics. However, we decided to use the Genius API where we had to agree to terms of service. Given our project will not be deployed for commercial use, we believe it will fall under fair use for the famous artists (a grey area) and are completely free to use in regards to the FMA artists.

**Division of labor**: Briefly outline who will be responsible for which part(s) of the project.

* Man-Fang Liang: preprocessing FMA, stable diffusion model
* Alzahra Fayie: preprocessing MSD and web scraping, stable diffusion model
* Qien Lin: audio to spectrogram conversions, stable diffusion model
* Zetao Wu: Transformer model design, stable diffusion model