Project Proposal: Deep Music Source Separation

Winter 2026 Final Project

Team Proposal

1 Executive Summary

Instead of selecting one of the three pre-defined options (Pose Estimation, Hallucinations, Inpainting), we propose an **"Alternative Project"** focused on **Music Source Separation (MSS)**.

The Goal: To take a mixed music track (a "mixture") and use Deep Learning to extract a single instrument stem (e.g., piano) from it.

The Research Angle: We will implement and compare two distinct deep learning paradigms:

1. **Frequency Domain (The "Image" Approach):** Treating audio as a 2D Spectrogram and using computer vision techniques (U-Net).
2. **Time Domain (The "Waveform" Approach):** Treating audio as a 1D signal and using raw audio convolutions (Demucs architecture).

This project fulfills the course requirement to "train a model... working with a non-trivial amount of data" and extends learned material in a "research context".

2 Methodology & Architectures

We will implement two different models to solve the same problem, allowing us to perform the comparative analysis required for a high-grade report.

2.1 Approach A: Frequency Domain (Baseline)

- **Concept:** We convert the audio waveform into a **Spectrogram** using the Short-Time Fourier Transform (STFT). This turns the audio problem into an image segmentation problem.
- **Architecture: 2D U-Net.**
  - **Input:** A spectrogram of the noisy mixture ($Time \times Frequency$).
  - **Output:** A "soft mask" (values 0 to 1) that we multiply element-wise with the input to isolate the instrument.
- **Pros:** Easier to visualize; utilizes standard CNN architectures.
- **Cons:** STFT discards phase information. Reconstructing the audio often results in artifacts because the phase must be approximated.

2.2 Approach B: Time Domain (Main Contribution)

- **Concept:** We feed the raw 1D waveform directly into the network. This avoids the phase reconstruction issue entirely.
- **Architecture: Simplified Demucs.**
  - **Input:** Raw tensor of audio samples (e.g., shape `[Batch, Channels, Time]`).
  - **Technique:** Uses **1D Convolutions** with a specific stride to downsample the audio, effectively learning its own frequency decomposition (an "incentivized DFT"). It utilizes an **LSTM (RNN)** in the bottleneck to capture long-term context.
- **Pros:** State-of-the-Art (SOTA) performance; handles phase perfectly.

3 Dataset & Training Strategy

We will use **Supervised Learning**, which creates perfect "Ground Truth" data efficiently.

- **Dataset: MUSDB18.** This is the standard academic dataset for this task. It contains ∼150 tracks where the stems (Vocals, Drums, Bass, Other) are stored as separate files.
- **Data Generation:** We do **not** need to label data manually. We load a "Piano" track and a "Drums" track, sum them together to create a mixture X, and task the model with predicting the original Piano track Y.
- **Loss Function:** L1 Loss (Mean Absolute Error) between the predicted waveform and the target waveform.

4 Work Plan & Report Structure

The course guidelines state that alternative projects are "not recommended" unless the students are "highly driven" and provide a detailed "Related Work" section.

Proposed Report Sections:

1. **Introduction:** Define the "Cocktail Party Problem" and the shift from Spectral to Waveform modeling.
2. **Related Work:** Review key papers (Open-Unmix for spectral, Wave-U-Net for 1D convs, Demucs for SOTA).
3. **Methodology:** Detail our implementation of the 1D Convolution blocks and the U-Net.
4. **Experiments:**
   - **Quantitative:** Measure Signal-to-Distortion Ratio (SDR) on a held-out test set.
   - **Qualitative:** Visual comparison of Spectrograms and listening tests.
5. **Work Report:** Documentation of code written and challenges faced.

5 Feasibility & Logistics

- **Submission Deadline:** 5.3.2026.
- **Libraries:** PyTorch (core), `musdb` (data loading), `torchaudio` (transforms).

- **Compute Strategy:** Audio models can be memory-intensive. We will likely downsample audio to 16kHz or 22kHz to speed up training on standard GPUs.
- **Why this works:** It combines **Fourier Analysis** (STFT), **Linear Algebra** (Matrix operations), and **Deep Learning** (RNNs/CNNs), leveraging our existing academic strengths.