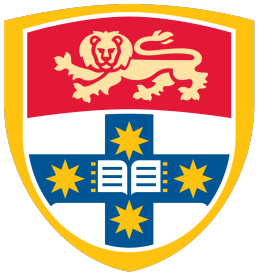
# ELEC 5305



THE UNIVERSITY OF SYDNEY

## GitHUb Link

Kaijun Cao SID:510621162

Faculty of Engineering
University of Sydney

**Date:** November 16, 2025

# Repository Link

The full codebase and experiment configuration are available at:

`https://github.com/Eric-Kaijun/ELEC5305-project-510621162-.git`

This appendix describes how to download the dataset, configure the environment, and run the training and evaluation pipeline.

# 1 Repository Setup

Clone the repository and enter the project directory:

```
git clone https://github.com/Eric-Kaijun/ELEC5305-project-510621162-.git
cd ELEC5305-project-510621162-
```

All experiments and figures in the report can be reproduced using `Code.ipynb` located at the repository root.

# 2 Dataset Preparation (NSynth)

The project uses the **NSynth** dataset, which contains more than 300k single-note musical instrument recordings at 16 kHz.

## Download

The dataset is available from the official Magenta project page:

- `https://magenta.tensorflow.org/datasets/nsynth`

Download the *JSON + WAV* version of the following splits:

- `nsynth-train`
- `nsynth-valid`
- `nsynth-test`

## Directory Structure

After extraction, your dataset directory should resemble:

```
/data/nsynth/
    nsynth-train/
    nsynth-valid/
    nsynth-test/
```

Manifest files such as `manifest_train_fs_fixed.csv` already included in the repository index the NSynth audio files. If your dataset path differs, simply update the variable `NSYNTH_ROOT` at the top of `Code.ipynb`.

# 3 Environment Configuration

A Python 3.9+ environment is recommended. Install dependencies with:

```
pip install numpy pandas matplotlib librosa
pip install torch torchvision torchaudio
pip install scikit-learn tqdm jupyter
```

For GPU acceleration, install a CUDA-compatible PyTorch build from the official PyTorch website.

# 4 Running the Full Pipeline

## Launching the Notebook

Start Jupyter in the repository directory:

```
jupyter notebook
```

Open the file `Code.ipynb` from the interface.

## Configuring Dataset Paths

Set your NSynth directory, for example:

```
NSYNTH_ROOT = "/data/nsynth"
```

Ensure that required files: `manifest_*.csv`, `feature_stats.npz`, and `label2idx_fs.json` remain in the repository root.

## Executing the Notebook

The notebook consists of seven stages:

1. Dataset indexing (load manifests and construct splits)
2. Feature extraction (STFT and CQT with global normalisation)
3. Augmentation (polyphonic mixing and signal degradation)
4. Dual-branch encoder (STFT and CQT branches)
5. Cross-view fusion (attention + FiLM modulation)
6. Multi-task training (instrument, family, octave, velocity)
7. Evaluation (metrics, confusion matrices, qualitative examples)

Run the cells sequentially to reproduce the results in the report.

# 5 Evaluation Only Using `last.pt`

To evaluate the model without re-training, use the saved checkpoint `last.pt`.

### Steps

1. Open `Code.ipynb`.
2. Locate the section titled: "Step-7 Evaluation Script (ONLY evaluate last.pt)".
3. Set the checkpoint path:

```
from pathlib import Path
CKPT_PATH = Path("last.pt")
```

4. Execute all evaluation cells.

This will compute test-set accuracy, per-class scores, the confusion matrix, and spectral visualisations for selected examples.

# 6 Repository Structure

- **Code.ipynb** – full preprocessing, training, evaluation pipeline
- **last.pt** – trained multi-view, multi-task model
- **manifest_train_fs_fixed.csv** – training manifest
- **manifest_val_fs_fixed.csv** – validation manifest
- **manifest_test_fs_fixed.csv** – test manifest
- **feature_stats.npz** – STFT/CQT normalisation statistics
- **label2idx_fs.json** – mapping from labels to indices
- **timbre_attrs.csv** – auxiliary timbre attributes

# 7 Summary

This repository provides a fully reproducible implementation of the multi-view, multi-task timbre classification framework. Following the steps above, users can:

- prepare the NSynth dataset,
- configure the environment,
- train the STFT–CQT model from scratch, and
- evaluate the final system using `last.pt`.