

PyNeuralFx: A Python Package for Neural Audio Effect Modeling

Yen-Tung Yeh¹, Wen-Yi Hsiao², Yi-Hsuan Yang¹

¹National Taiwan University, ²Independent Researcher



Overview

- Easy-to-use framework for neural audio effect modeling
- Provided several tools for easily experiment setup
 - Loss functions
 - Evaluation metrics
 - Visualization

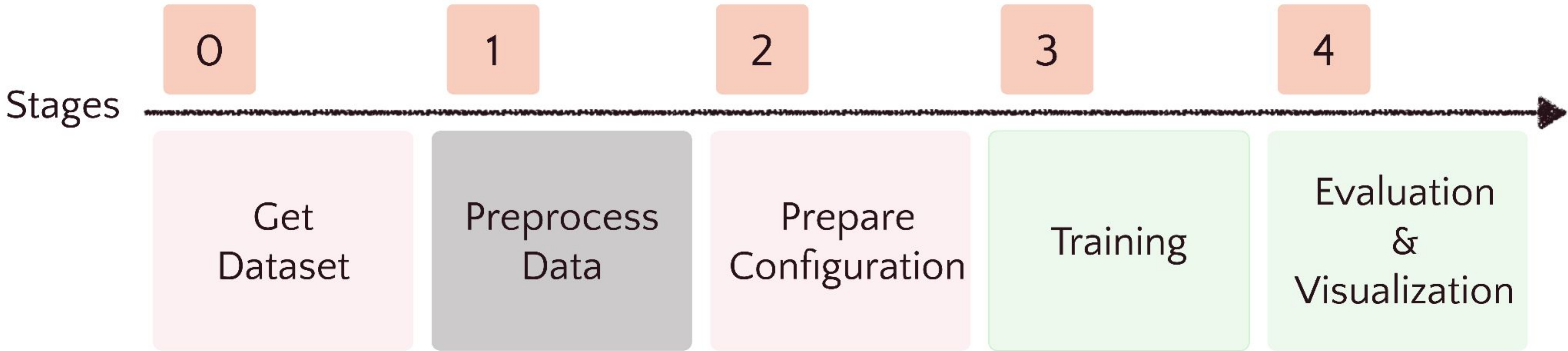
Arxiv:

GitHub:

PyNeuralFx



Workflow



0

Get Dataset

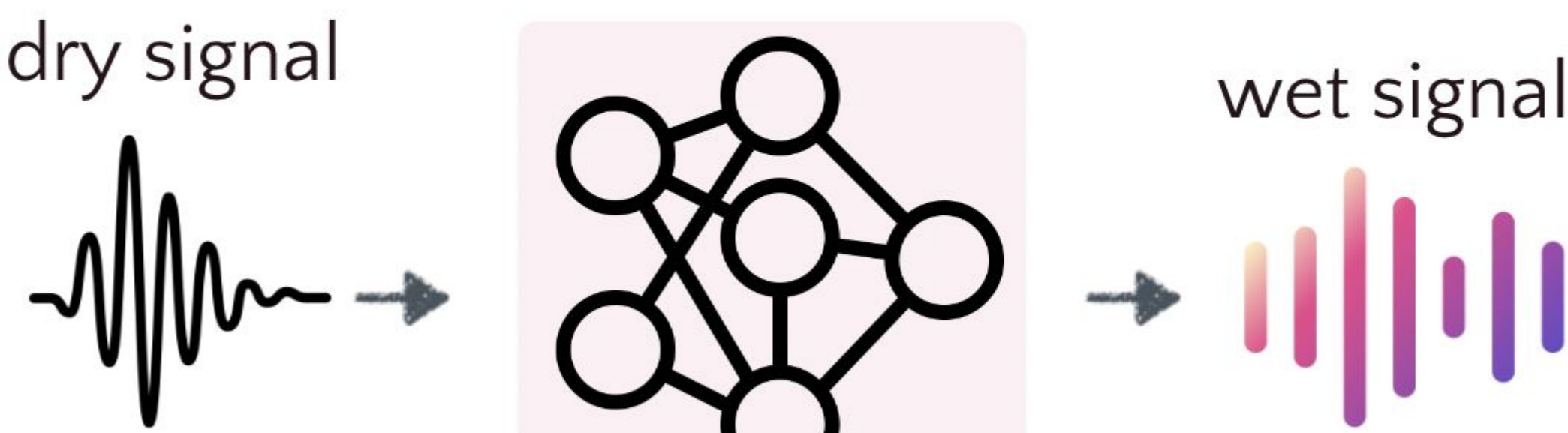
- Download dataset from common academic dataset
- Prepare your own data
 - Using audio-plugin to render effects
 - Collect output from analog-device

Notice:

- Sample rate is important!
- Check your data carefully, make sure the high-quality data
- Know the difference between **snapshot** and **full modeling**

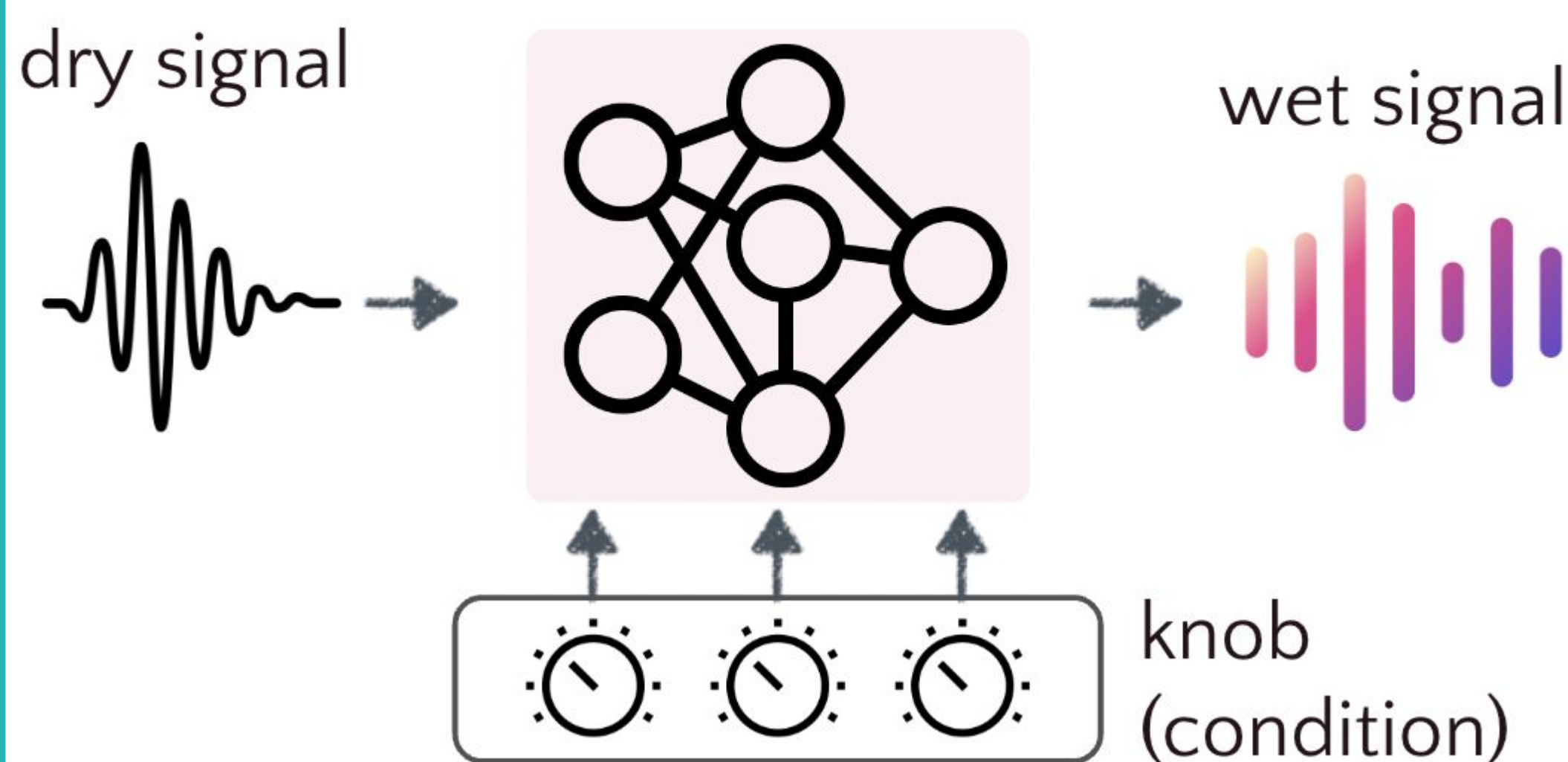
Snapshot

neural network



Full Modeling

neural network



Preprocess Data

- Different datasets often have different data formats. To use our framework, we define the specific rule for data format (including naming and structure)
- **Writing codes** to match the naming and structure 🤖
- However, it is always fine to write your own dataloader, our framework is easy to modify

2

Prepare Config

- Prepare settings for experiments
- Model Arch:
 - Snapshot modeling:
 - snapshot-tcn
 - snapshot-gcn
 - snapshot-vanilla-rnn
 - snapshot-lstm
 - snapshot-gru
 - Full modeling:
 - Concat-{tcn, gcn, rnn, lstm, gru}
 - FiLM-{tcn, gcn, rnn, lstm, gru}
 - Hyper-{tcn, gcn, rnn, lstm, gru}

2

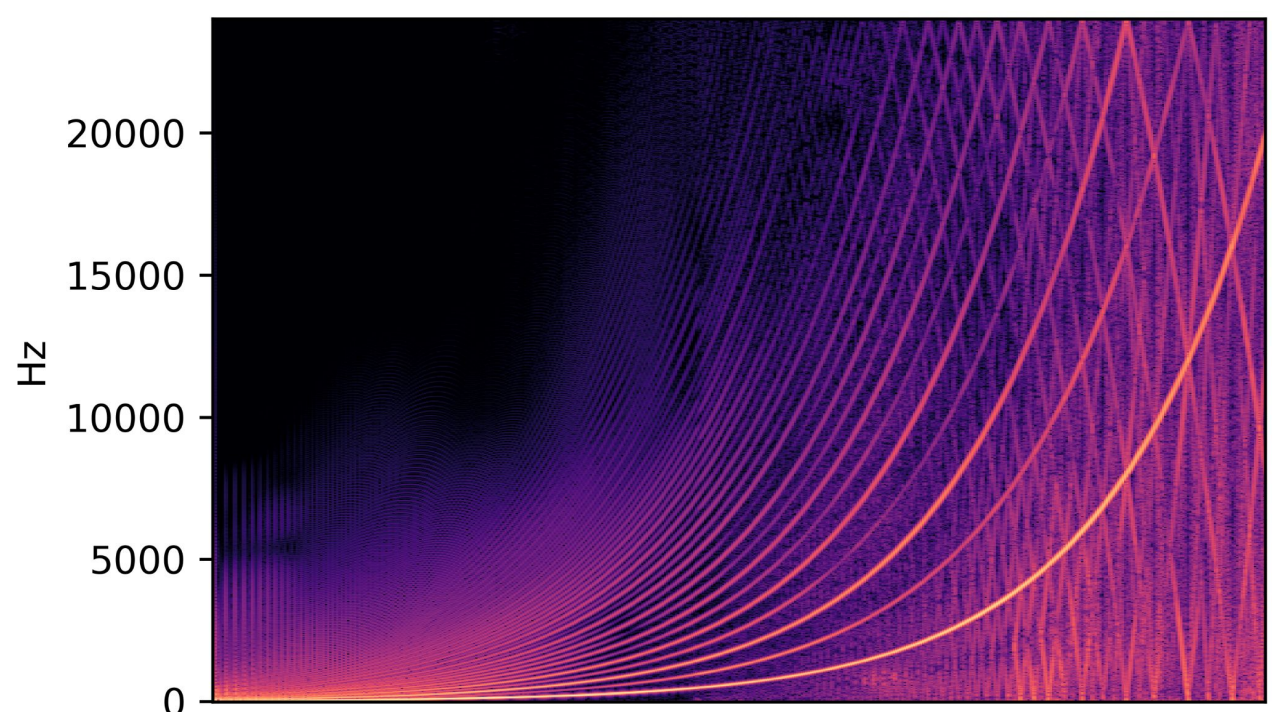
Training

Run the training code and wait for your neural network to emulate audio effects

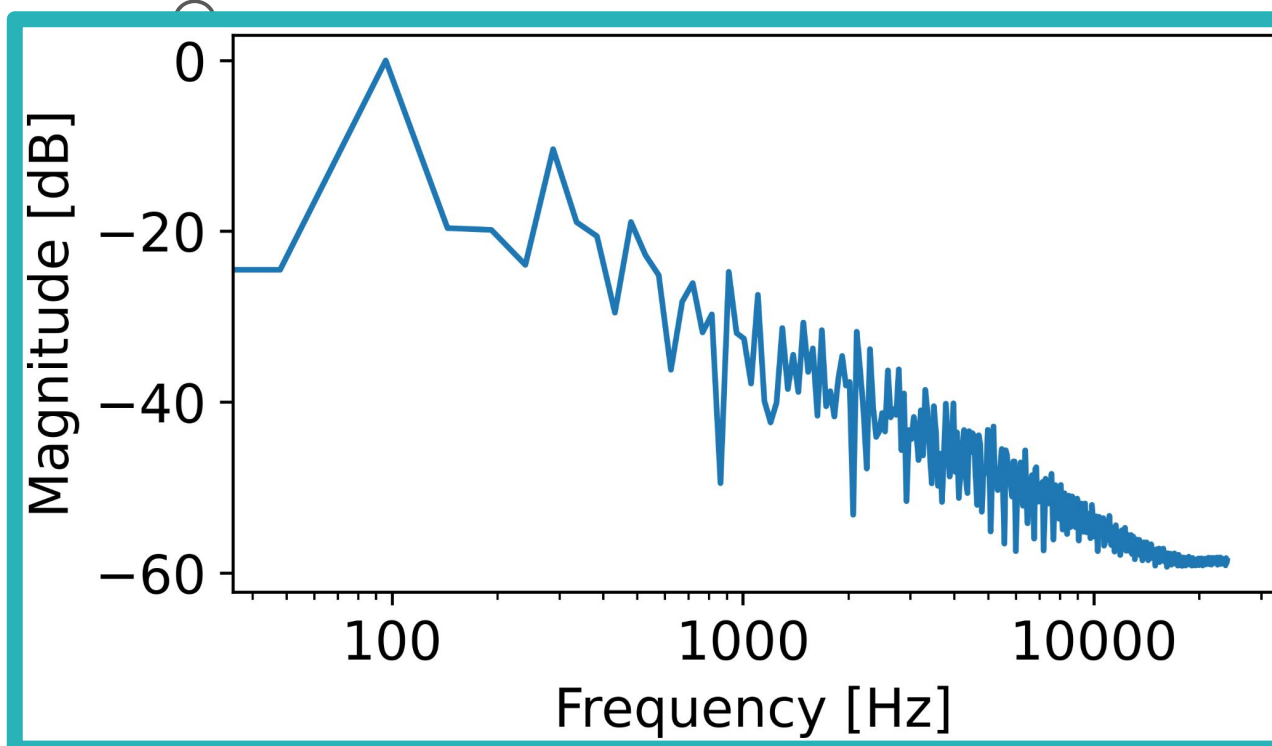


Evaluation

- Choose the proper metrics we provided
 - Reconstruction loss
 - Time-domain
 - Freq-domain
 - Loudness error
 - LUFS
 - Transient Metric
 - Audio Feature
 - RMS
 - Spectral Centroid
- Using visualization tools
 - Aliasing in neural networks



Harmonic response of trained neural networks



Collaboration

Feel free to contact me for improving the package. I plan to include more architectures, such as DDSP-based architectures and more evaluation metrics