Digital Guitar Distortion

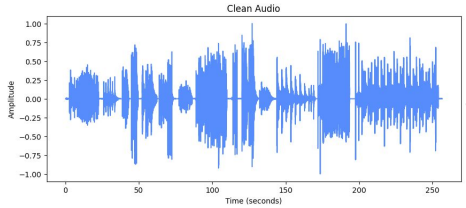
using Recurrent NNs

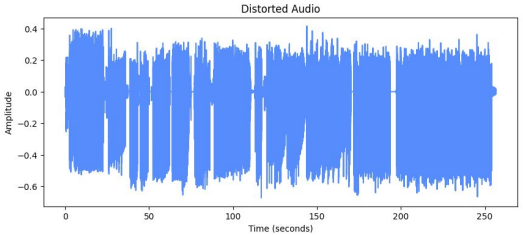
Jacob, Martin, Nikolaj, Patrick

The why?

-Motivation

- Effects such as **distortion**, overdrive, reverb, flanger... are **ubiquitous in modern music**.
- **Traditionally achieved using analog equipment.** Compression from tape, distortion from signal clipping in preamps
- **Music has gone digital** using digital audio workstations (Daw's).
- **We want to emulate the analog effects using machine learning** to provide an easy way to achieve these in a digital environment

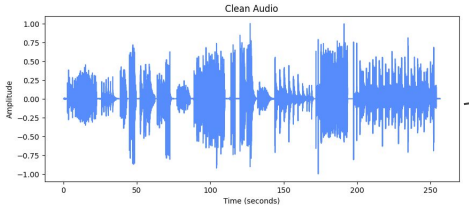




The why?

-Motivation

- Effects such as **distortion**, overdrive, reverb, flanger... are **ubiquitous in modern music**.
- **Traditionally achieved using analog equipment.** Compression from tape, distortion from signal clipping in preamps
- **Music has gone digital** using digital audio workstations (Daw's).
- **We want to emulate the analog effects using machine learning** to provide an easy way to achieve these in a digital environment



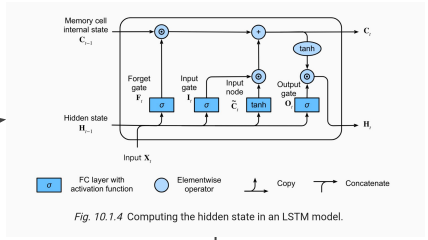
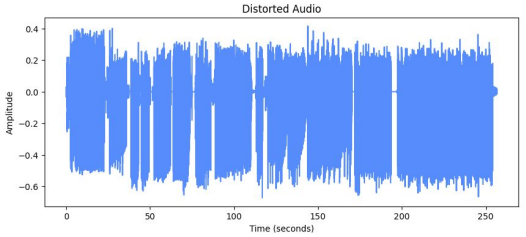
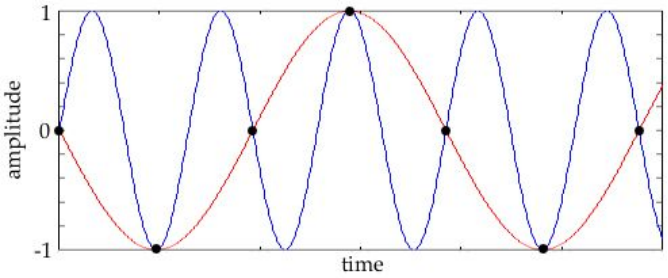


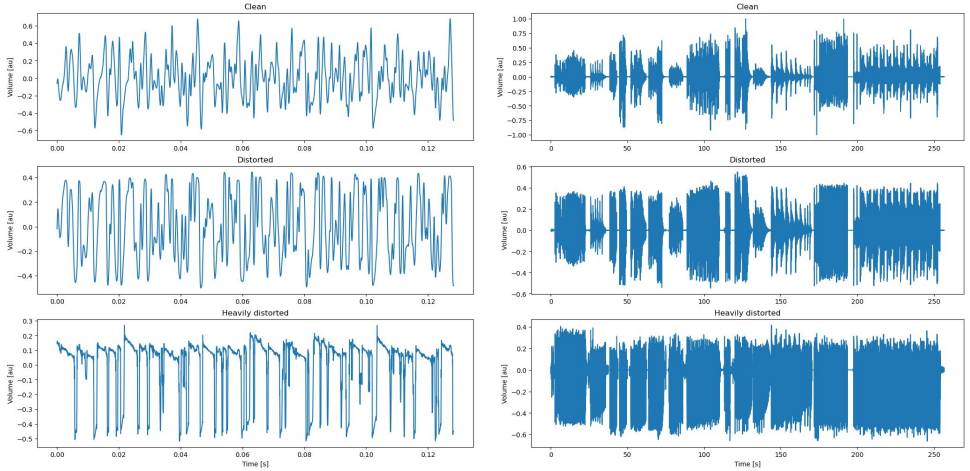
Fig. 10.1.4 Computing the hidden state in an LSTM model.



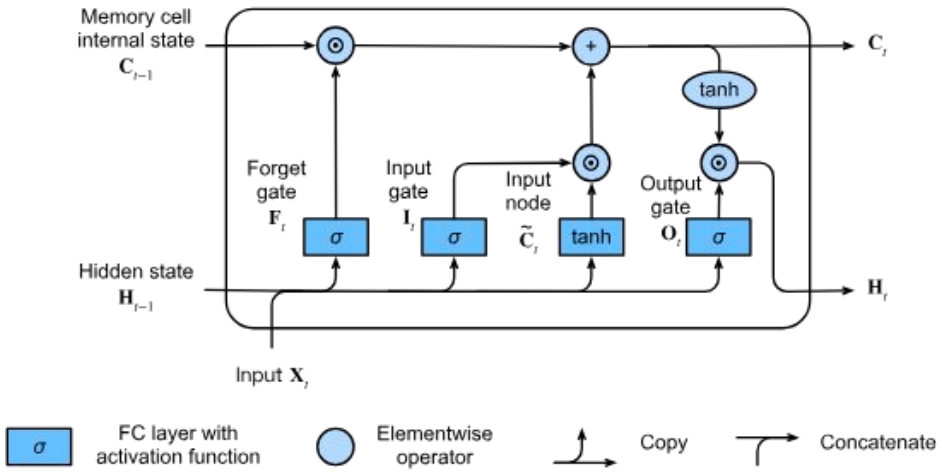
Data

- ~11 million samples per file with each their volume (i.e. an array of floats with a length of 11 million)
- .wav files of guitar playing at different levels of distortion
- Sample rate of 44100 Hz (2x Nyquist frequency) ensures all audible frequencies are represented





LSTM - Long Short-Term Memory

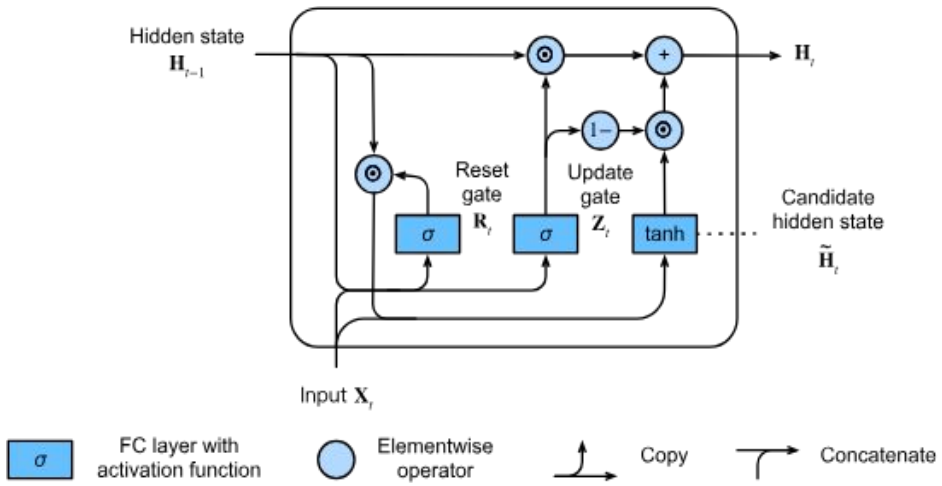


Passes two states and one input through three gates:

1. How much of previous long-term memory do we forget?
2. How should current input and short-term memory affect the long-term memory?
3. How should current long-term memory affect the short-term memory?

Dive Into Deep Learning (https://d2l.ai/chapter_recurrent-modern/index.html)

GRU - Gated Recurrent Unit



Passes one state and one input through two gates:

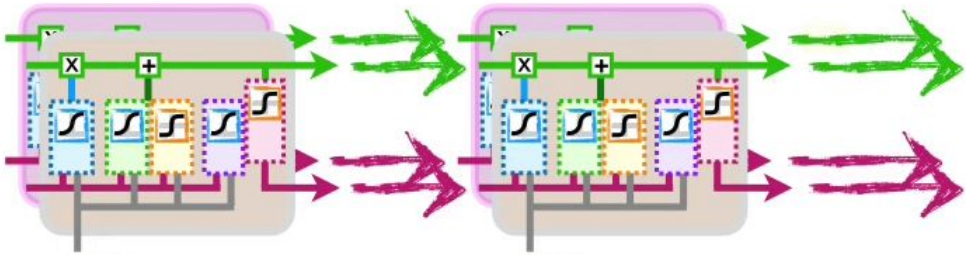
1. How much of the previous memory should be used for the current memory update?

2. How much of the previous memory should be forgotten? The more we forget, the more we'll use from our current memory update

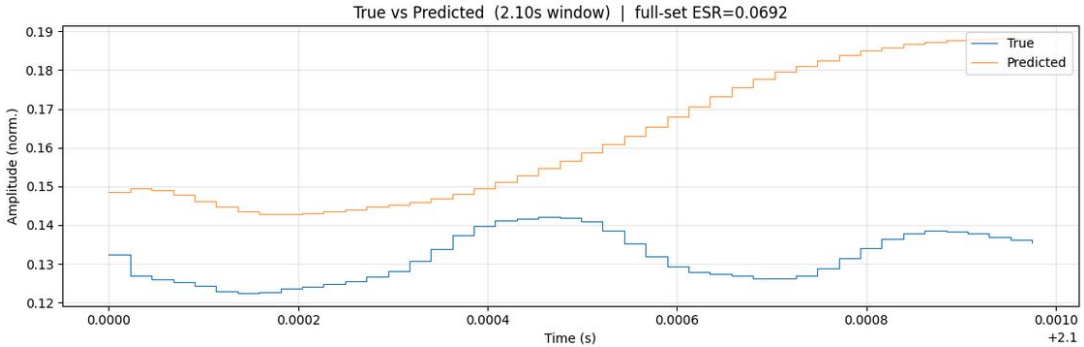
Dive Into Deep Learning (https://d2l.ai/chapter_recurrent-modern/index.html)

Our models

- Two simple models using GRU and LSTM
- HPs tuned:
  - LR
  - hidden layers
  - seq length
  - weight decay (overfitting)
- One NN + GRU (preprocessing)

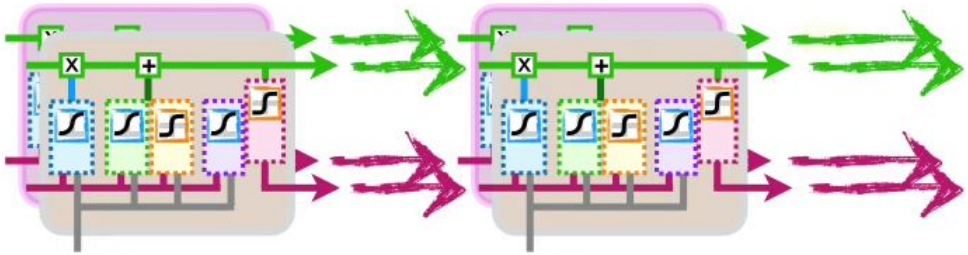


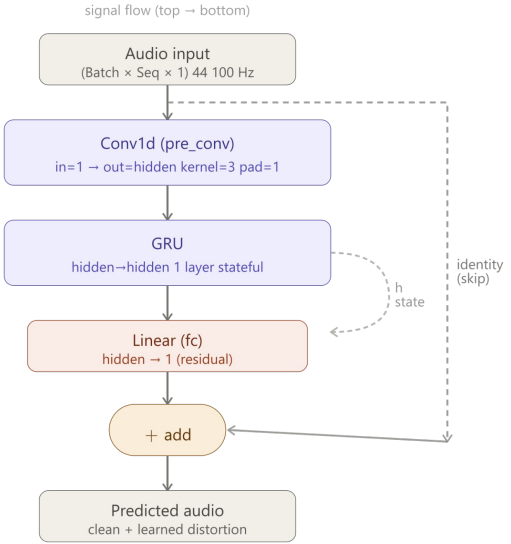
Source: StatQuest on Youtube



Our models

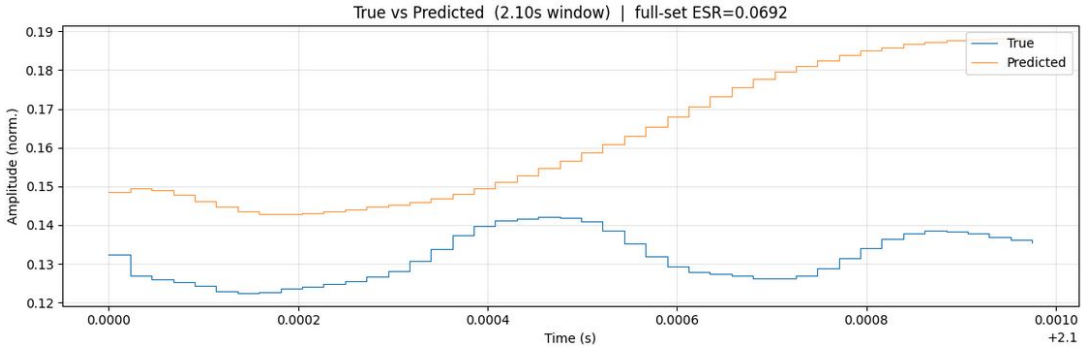
- Two simple models using GRU and LSTM
- HPs tuned:
  - LR
  - hidden layers
  - seq length
  - weight decay (overfitting)
- One NN + GRU (preprocessing)



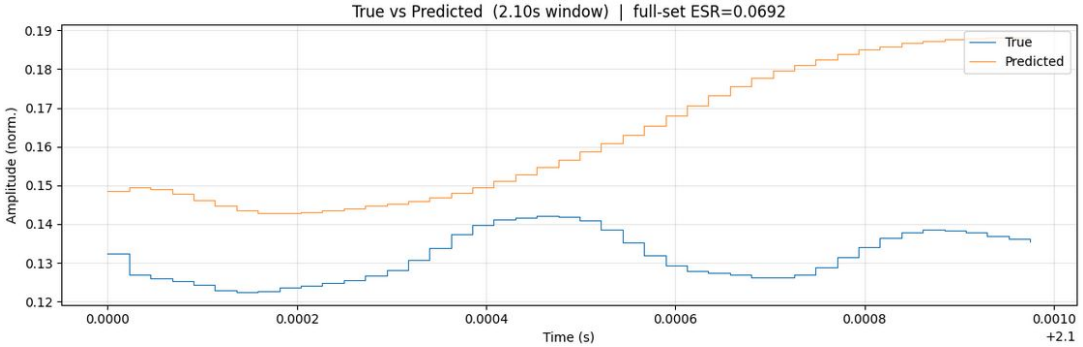


Optimization

- Training samples = seq len * batches
- 20x Bayesian HP tuning done with **median pruning**
- Final HP tuning done with **patience**
- Training / val / test - 80/10/10 (intentionally no cross validation).



Optimal Loss



$$ESR = \frac{\sum \Delta S}{\sum S}$$

$$Emph = S[i+1] - \alpha S[i]$$

$$Emph = \alpha(S[i+1] - S[i]) + (1-\alpha)S[i]$$

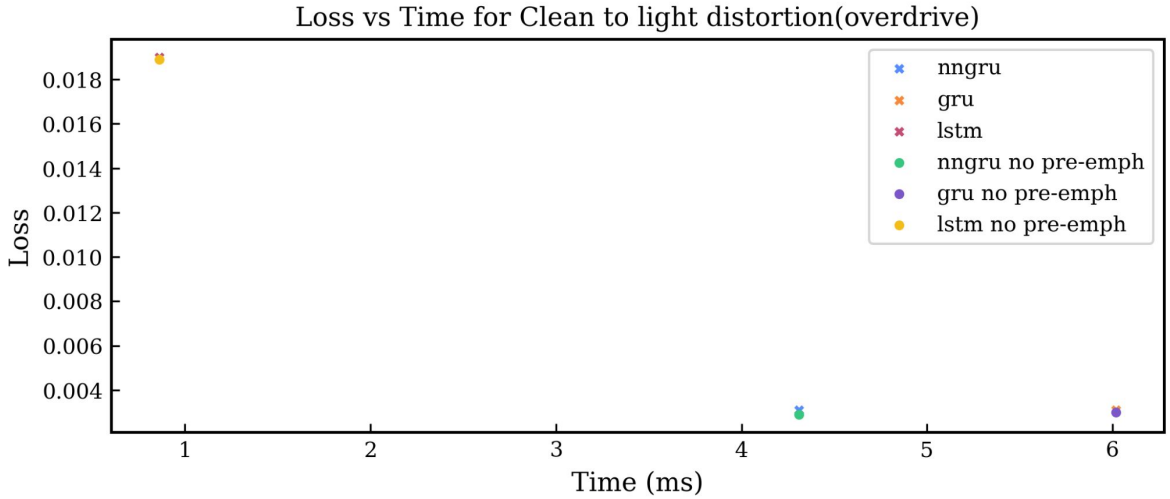
Emph_s = s[:, 1:, :] - a s[:, :-1, :]= (1-a) s[:, 1:, :] + a (s[:, 1:, :] - s[:, :-1, :])

Results

- **GRU models achieve x4 times lower loss** compared to Lstm
- **All models are within the speed bound** given by the sequence length.

$$t_{max} = \frac{512}{44100} \approx 11ms$$

- We chose to move forward with the two GRU based models
- *Low process time of lstm is weird (LSTM and GRU have approximately the same hyperparameters)*



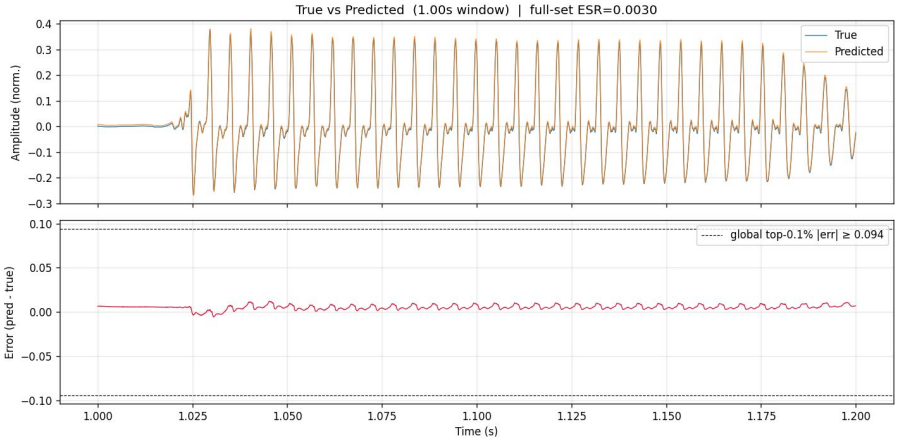
Results

- **GRU models achieve x4 times lower loss** compared to Lstm
- **All models are within the speed bound** given by the sequence length.

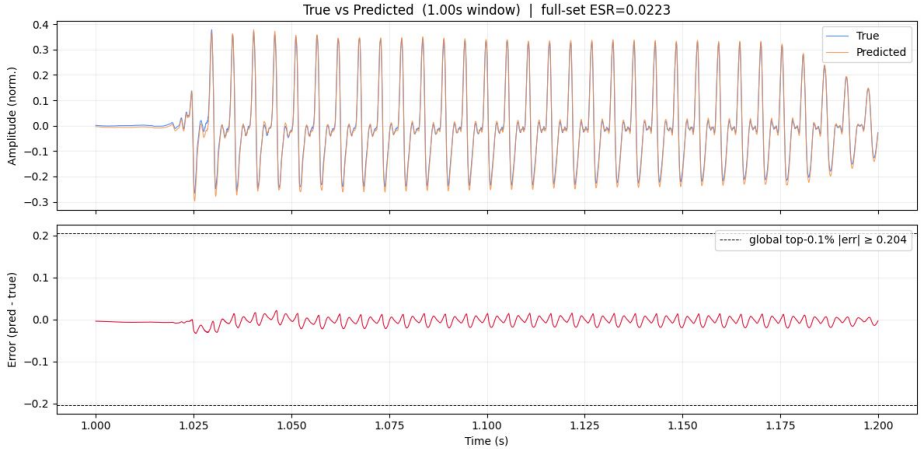
$$t_{max} = \frac{512}{44100} \approx 11ms$$

- We chose to move forward with the two GRU based models

GRU



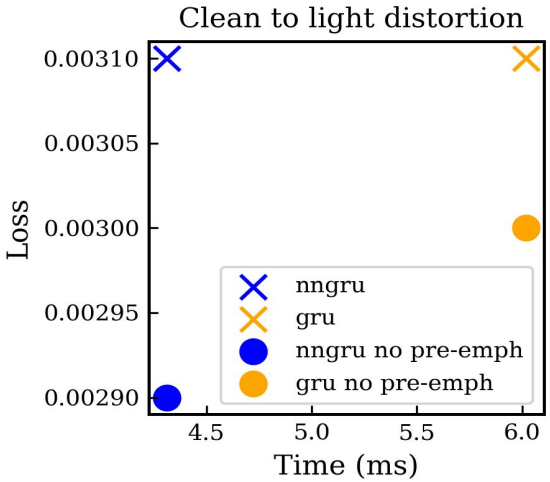
LSTM

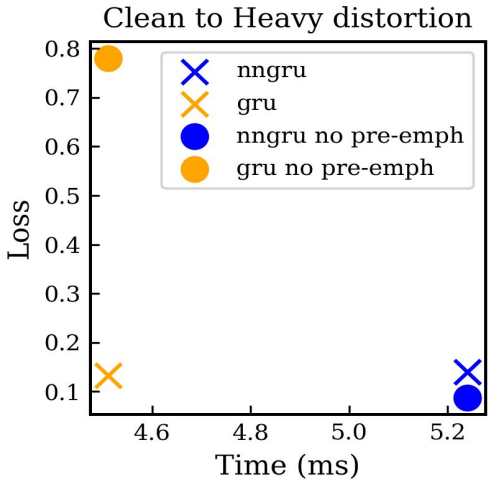


Results

- Larger loss with heavier distortion
- Pre emphasis have a higer loss but might produce a “better sounding” model.
- Both models produce convincing emulations.

- Input/validation sound





Results

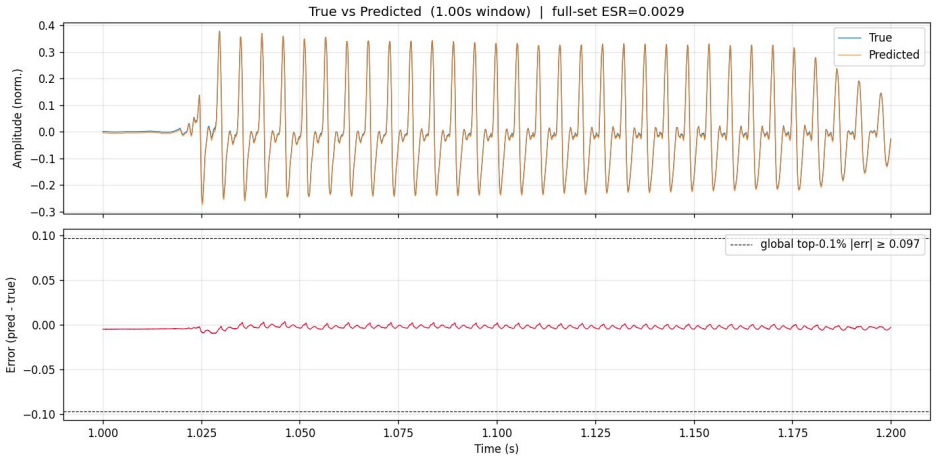
- Larger loss with heavier distortion
- Almost perfect first for the light distortion

- Input/validation sound

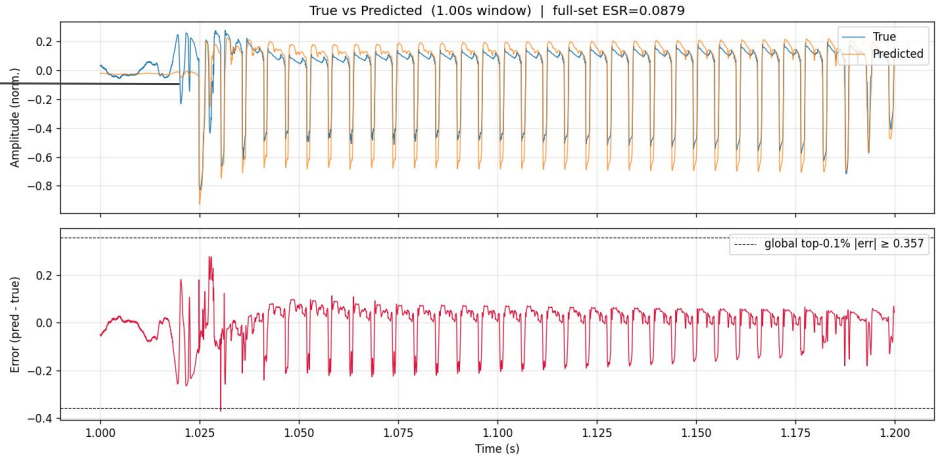
The transient is hard to model

This might be a consequence of an inertia in the hidden state

Light distortion

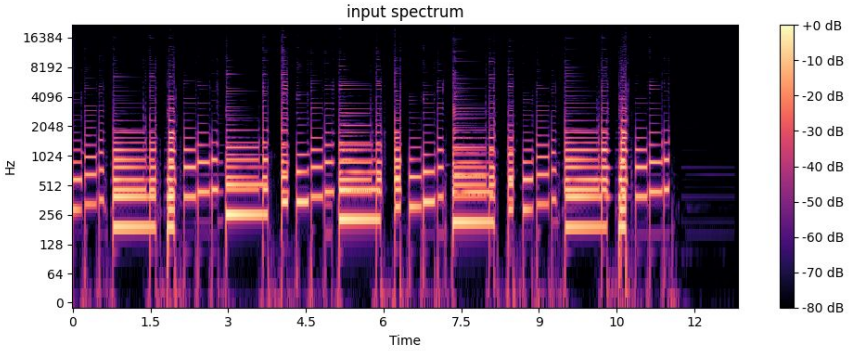


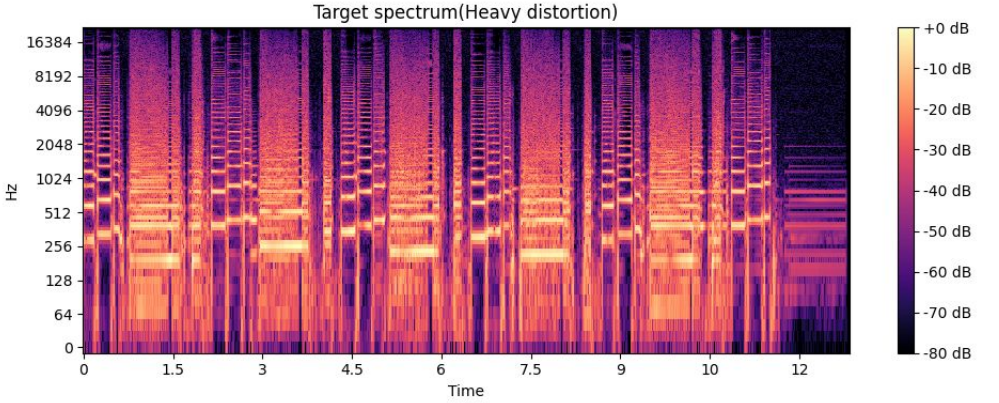
Heavy distortion

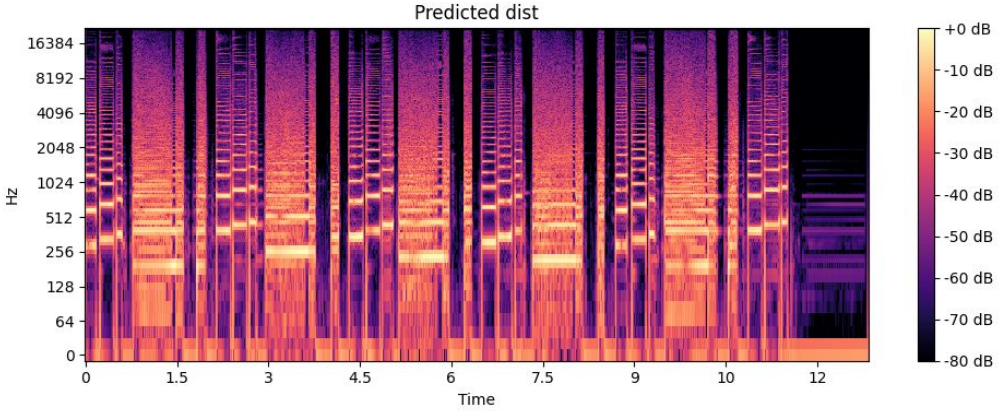


Results

- Power spectrum analysis
- Similar amount of energy in target and predicted spectra.
- Mismatch in low freq (50hz) is not as important for the instrument







Live Processing

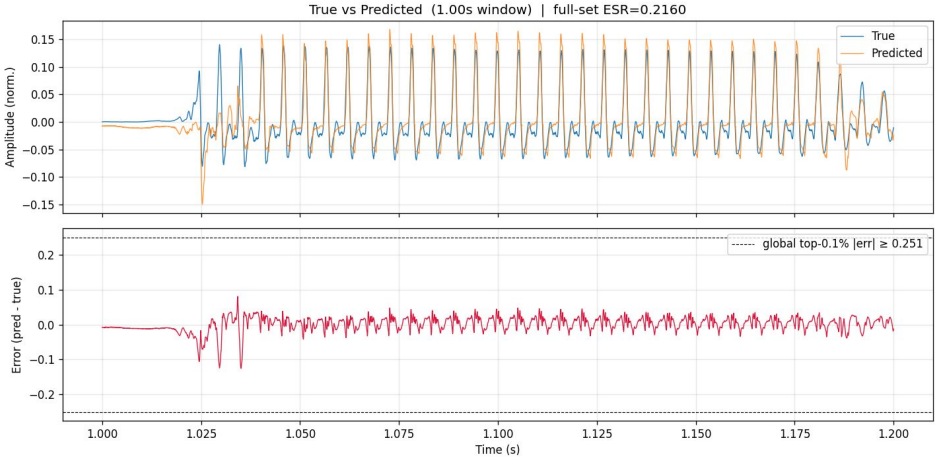
- Export the Pytoch model to c++ via the TorchLib format
- Real time processing via the VST interface
- Built using the JUCE framework
- Test in Ableton

What about cleaning a distorted guitar?

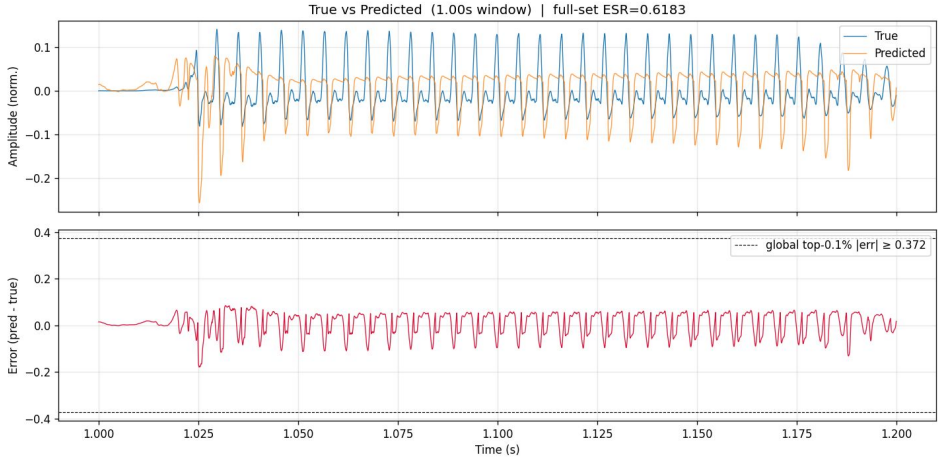
Cleaning distorted data

- Reversing input and target Data for both GRU based models
- Much higher loss, with predicted audio having audible distortion
- Works better for less distorted signals

NNGRU

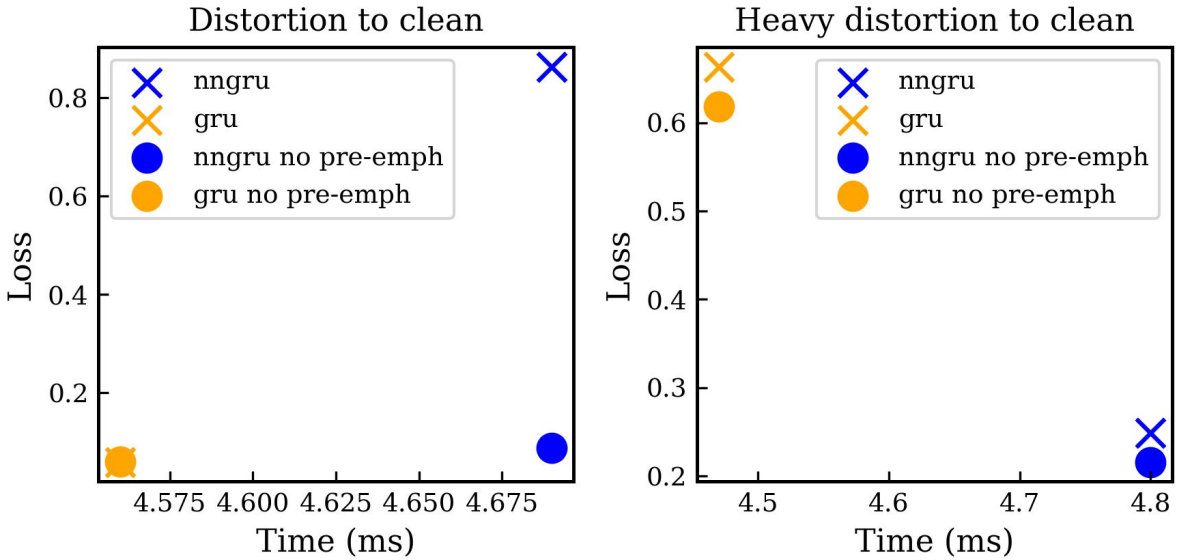


GRU



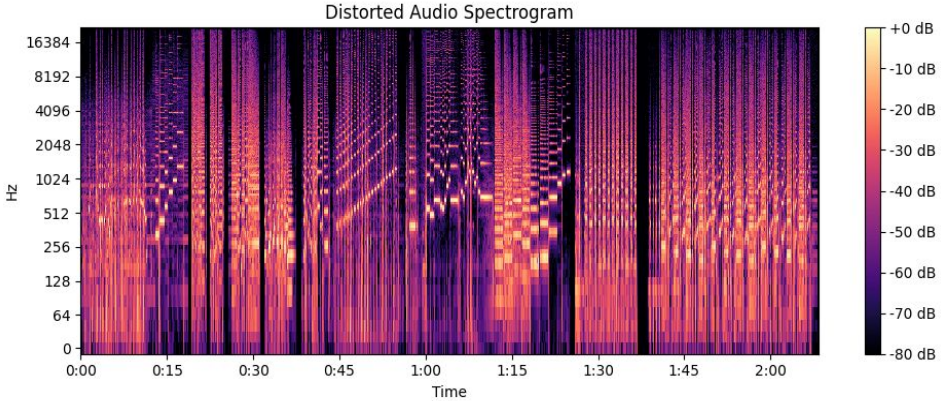
Cleaning distorted data

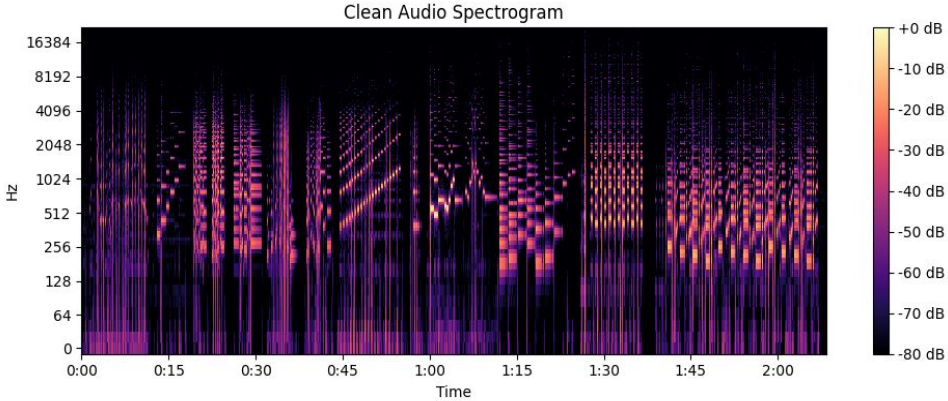
- Reversing input and target Data for both GRU based models
- **Generate a clean sample**, which together with the distorted would constitute a **new training set**



Cleaning distorted data

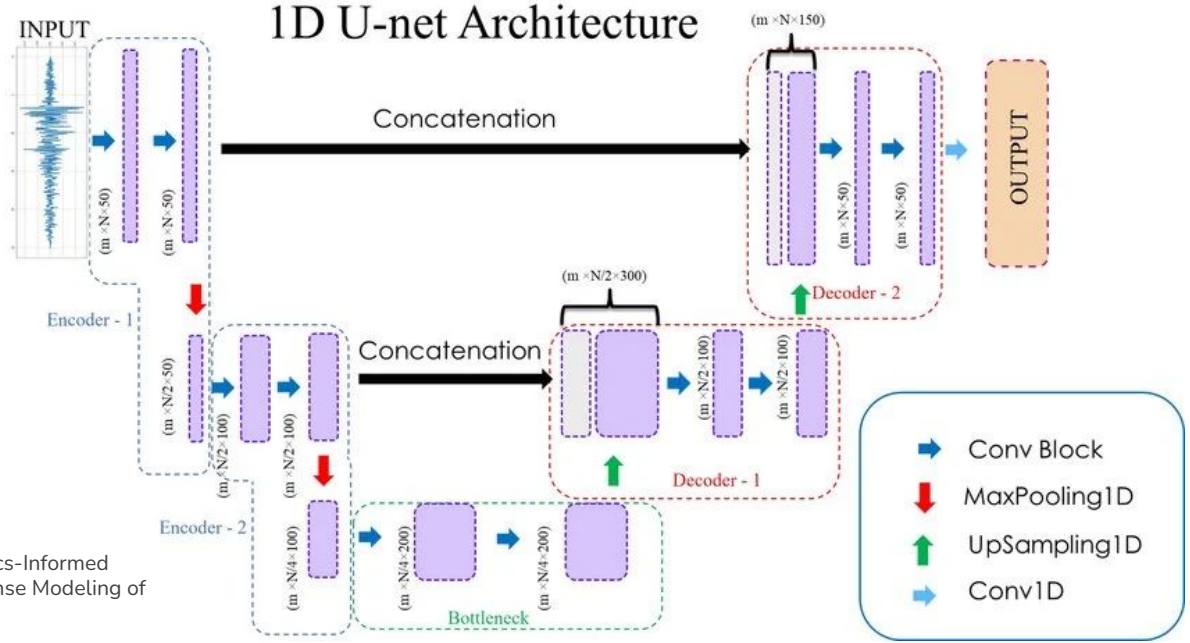
- Overdrive and distortion adds overtones which are visible in a power spectrum





DEMUCS - Deep Extractor for Music Sources

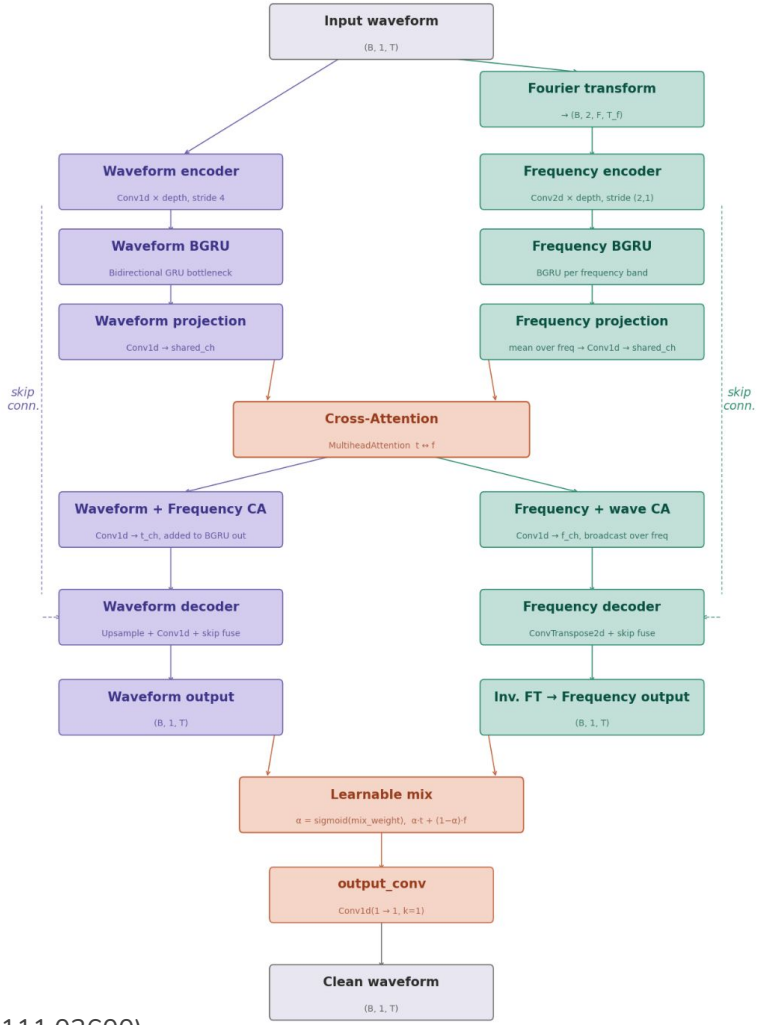
- Sony paper
- Source separation
- Different Layers
- More Channels
- Skip Connections



BISWAS, SUTIRTHA & Yadav, Kshitij. (2025). A Physics-Informed U-net-LSTM Network for Data-Driven Seismic Response Modeling of Structures. 10.2139/ssrn.5382270.

Our DEMUCS

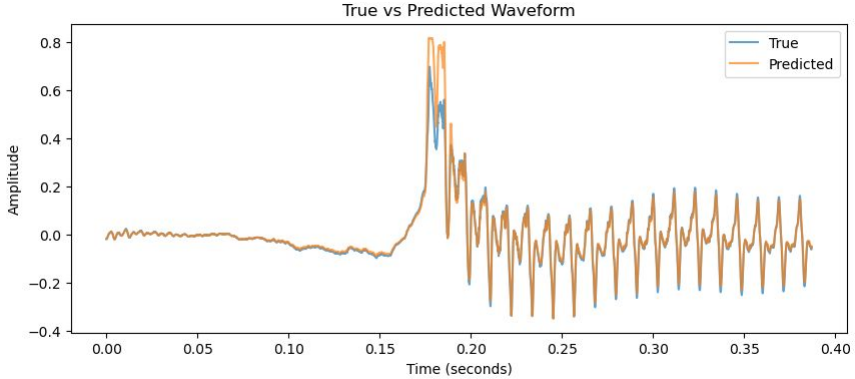
- Spectrogram representation makes distortion easy to identify
- Waveform preserves phase
- We use cross attention to exchange information between both domains

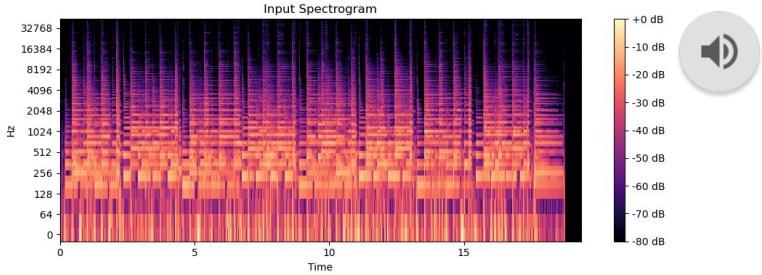


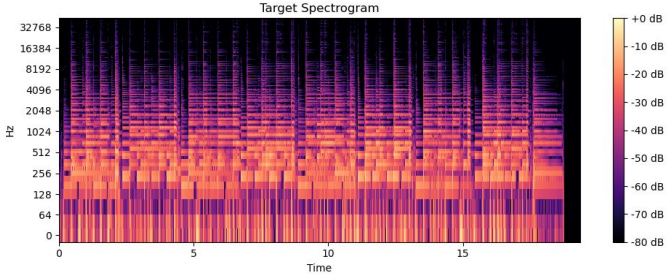
Défossez, Alexandre, *Hybrid Spectrogram and Waveform Source Separation* (2021) (arXiv:2111.03600)

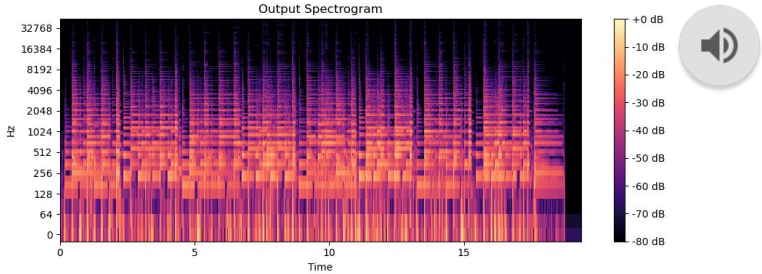
DEMUCS results

- Trained on ~50 minutes of data (~126 million samples)
- Optimization and training is time consuming but here's a proof of concept









Appendix

NN GRU

Plots were produced using these parameters

Clean to distorted


```
{
  "params": {
    "lr": 0.010299777367940863,
    "hidden": 48,
    "seq len": 512,
    "weight_decay": 1.3075815622867645e-06
  },
  "search esr": 0.008729771710932255,
  "n trials": 20,
  "fixed": {
    "sr": 44100,
    "batch size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```


Clean to heavily distorted


```
{
  "params": {
    "lr": 0.007456241820636362,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 2.0236454843648914e-06
  },
  "search_esr": 0.2593100070953369,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "kernel_size": 3,
    "pre_emph": 0.85
  }
}
```


Heavily distorted to clean


```
{
  "params": {
    "lr": 0.01206649984713325,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 1.0032003337722586e-06
  },
  "search_esr": 0.5263362526893616,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "kernel_size": 3,
    "pre_emph": 0.85
  }
}
```


Distorted to clean


```
{
  "params": {
    "lr": 0.007456241820636362,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 2.0236454843648914e-06
  },
  "search_esr": 0.1346566379070282,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "kernel_size": 3,
    "pre_emph": 0.85
  }
}
```

GRU

Plots were produced using these parameters

Clean to distorted


```
{
  "params": {
    "lr": 0.010299777367940863,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 1.3075815622867645e-06
  },
  "search_esr": 0.008729771710932255,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```


Clean to heavily distorted


```
{
  "params": {
    "lr": 0.008530468805437319,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 1.7451820529925351e-06
  },
  "search_esr": 0.257701575756073,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```


Heavily distorted to clean


```
{
  "params": {
    "lr": 0.018205555786732146,
    "hidden": 48,
    "seq_len": 256,
    "weight_decay": 2.1517667249729845e-05
  },
  "search_esr": 0.5858762860298157,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```


Distorted to clean


```
{
  "params": {
    "lr": 0.01020210845859495,
    "hidden": 48,
    "seq_len": 512,
    "weight_decay": 1.2329814744074788e-06
  },
  "search_esr": 0.12053848057985306,
  "n_trials": 20,
  "fixed": {
    "sr": 44100,
    "batch_size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```

LSTM

Plots were produced using these parameters

Clean to distorted


```
{
  "params": {
    "lr": 0.018205555786732146,
    "hidden": 48,
    "seq len": 256,
    "weight_decay": 2.1517667249729845e-05
  },
  "search esr": 0.04463794454932213,
  "n trials": 20,
  "fixed": {
    "sr": 44100,
    "batch size": 32,
    "layers": 1,
    "pre_emph": 0.85
  }
}
```

DEMUCS

Plots were produced using these parameters

Heavily distorted to clean


```
{
  "params": {
    "lr": 0.008182284403479165,
    "channels": 64,
    "time depth": 3,
    "freq_depth": 3,
    "n fft": 256,
    "batch_size": 256,
  },
  "fixed": {
    "sr": 44100,
    "time stride": 4,
    "hop length": 1024,
    "gru layers": 2,
    "transformer_heads": 4,

  }
}
```