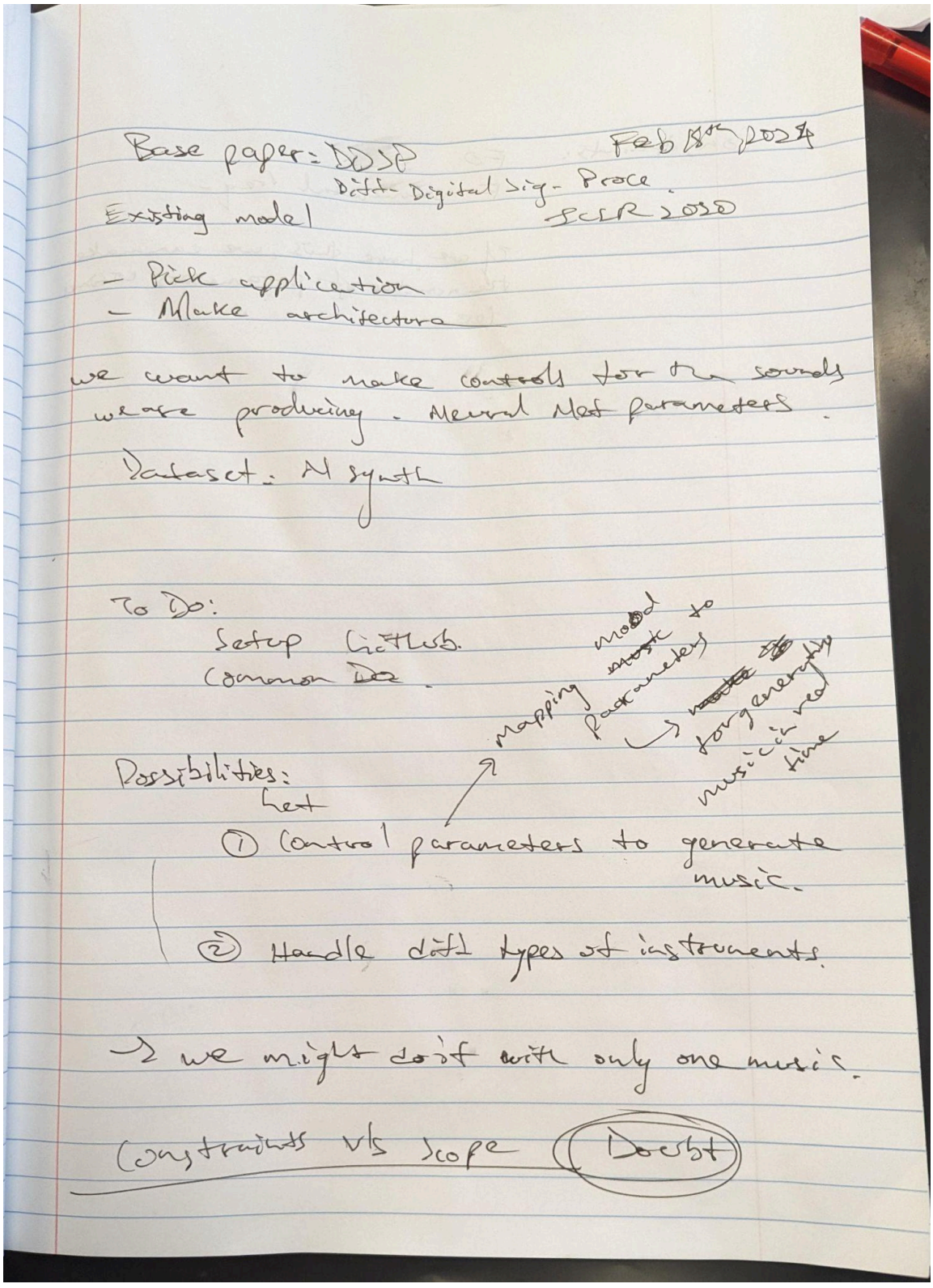
Notes:

You can add the notes/ details you want to share in the following pages. Everyone should have editor access.

meeting _1 [Feb 15th, 2024]



Base paper: DDSP — Feb 15th 2024
Diff. Digital Sig. Proce.
ICLR 2020

Existing model

- Pick application
- Make architecture

we want to make controls for the sounds we are producing. Neural Net parameters.

Dataset: NSynth

To Do:
Setup GitHub.
Common Doc.

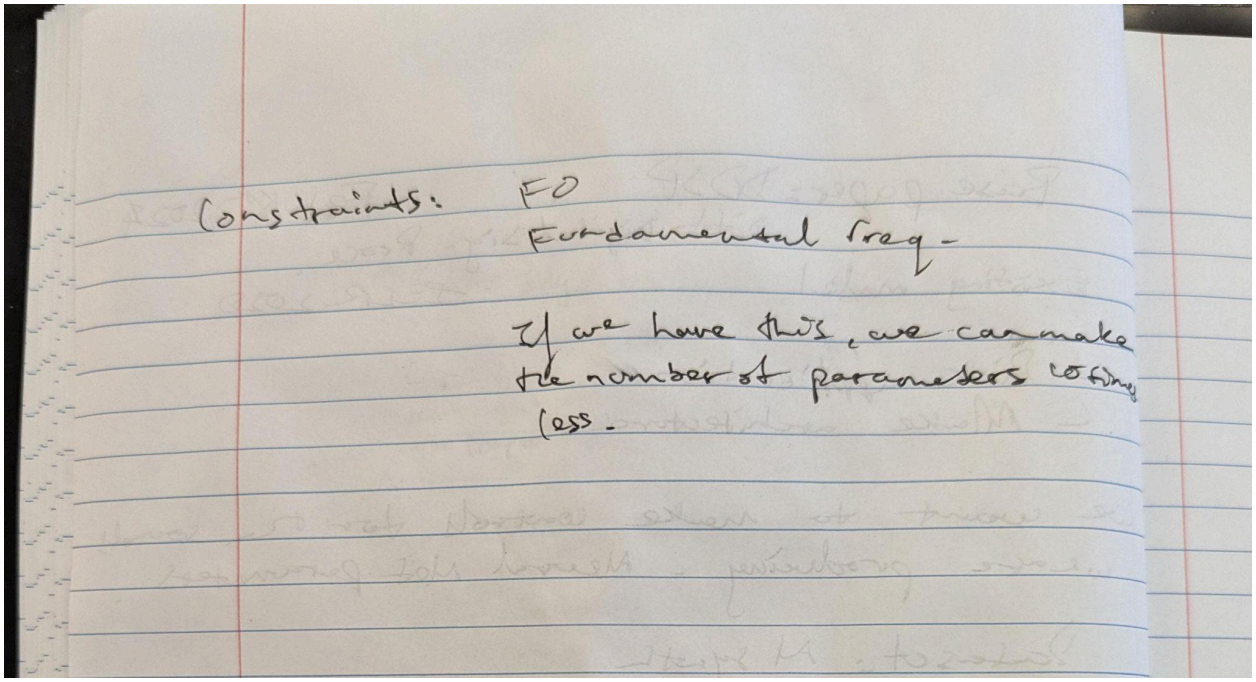
Possibilities:
Text

① Control parameters to generate music.
mapping mood to music parameters
for generating real-time music

② Handle diff types of instruments.

→ we might do it with only one music.

Constraints vs Scope — Doubt



Resources about DDSP

Papers and Tutorials

- A very good **DDSP tutorial**: Introduction to DDSP for Audio Synthesis
- The **original DDSP paper** (Engel et al. 2020): ICLR paper link
- A **detailed review** of DDSP in music and speech synthesis (Hayes et al. 2024): arXiv link, frontiers link
- A paper about **drum synthesis** using high-level timbre descriptors (brightness, depth, warmth) for control the synthesis (Lavault et al. 2022): paper link
- A paper about **find the synthesizer parameters** from an input audio signal (Masuda & Saito 2021): paper link
- A paper about **piano synthesis** by incorporating physical knowledge about the piano to the neural network design: paper link
- A paper about transfering the styles of audio effects from one recording to another (Steinmetz et al. 2022): paper link
    - a 44-minutes presentation video: YouTube link

Applications and Demos

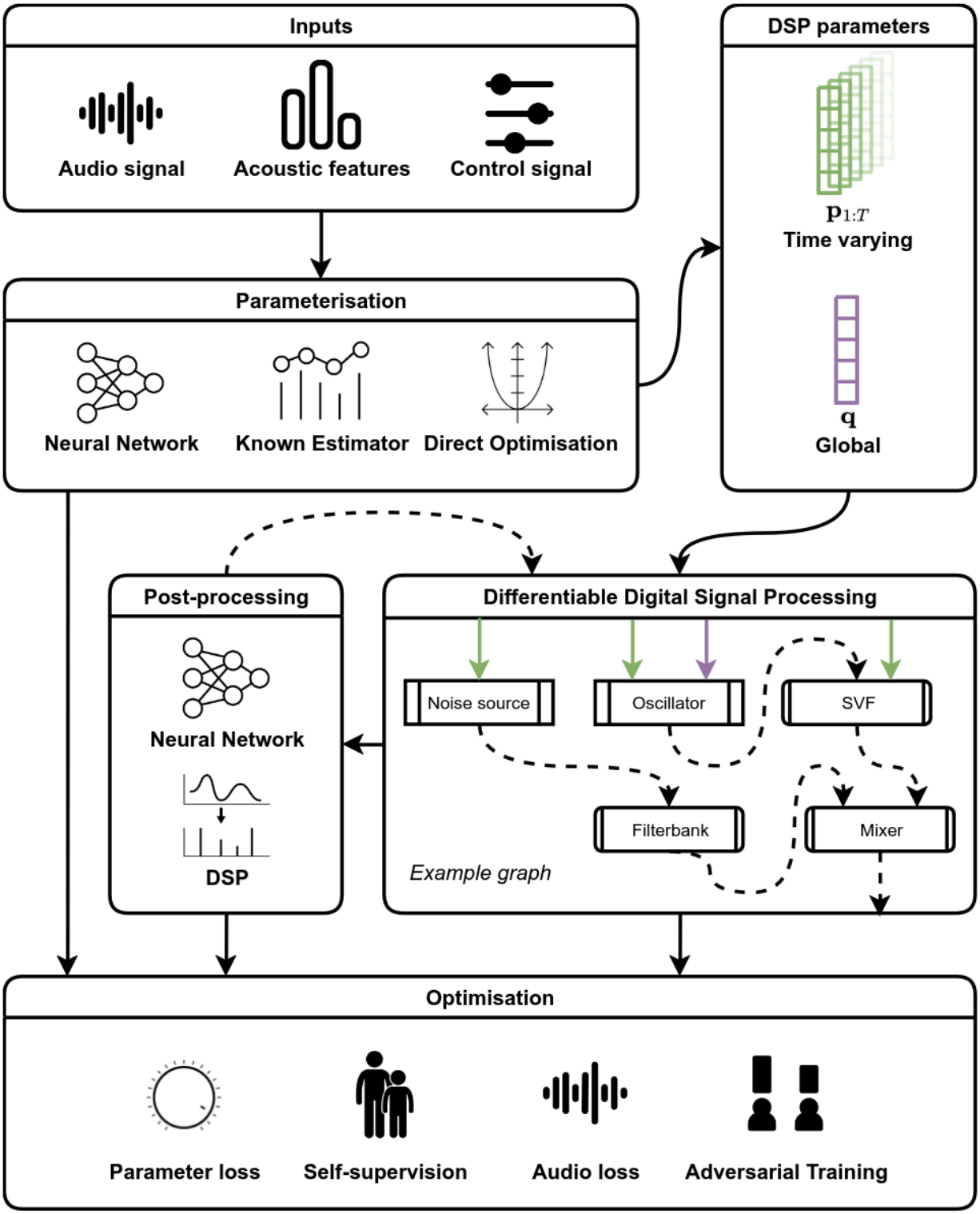
- Audio examples of the original DDSP paper: supplement link

- Another demo site about applying the original DDSP architecture directly for timbre transfer: Tone Transfer
- Examples of another architecture for neural synthesis: audio examples
    - high reconstruct quality and real-time ability by adding a GAN structure (Caillon & Esling 2021) paper link
- Sound examples for finding the synthesizer parameters and re-synthesis of the sounds: GitHub link
    - (Ye et al. 2023) paper link
- Sound examples for DDSP-Piano: supplement link
- Sound examples for "style transfer of audio effects" paper: GitHub link

Datasets

- NSynth (Engel et al. 2017): website
    - containing 305,979 musical notes
    - contains source, instrument family, qualities, pitch, and velocity
    - only 16kHz
- Amp-Space (Naradowsky 2021): paper link
    - large-scale dataset of paired audio samples
    - a source audio signal, and an output signal, the result of a timbre transformation
    - >500 hours synthesized audio + >50 hours real audio
    - 44.1kHz monaural sound
- synth1B1 (Turian et al. 2021): paper link
    - 1 billion 4-second synthesized sounds
    - paired with the synthesis parameters used to generate them
    - 44.1kHz

Two helpful pictures from the review paper



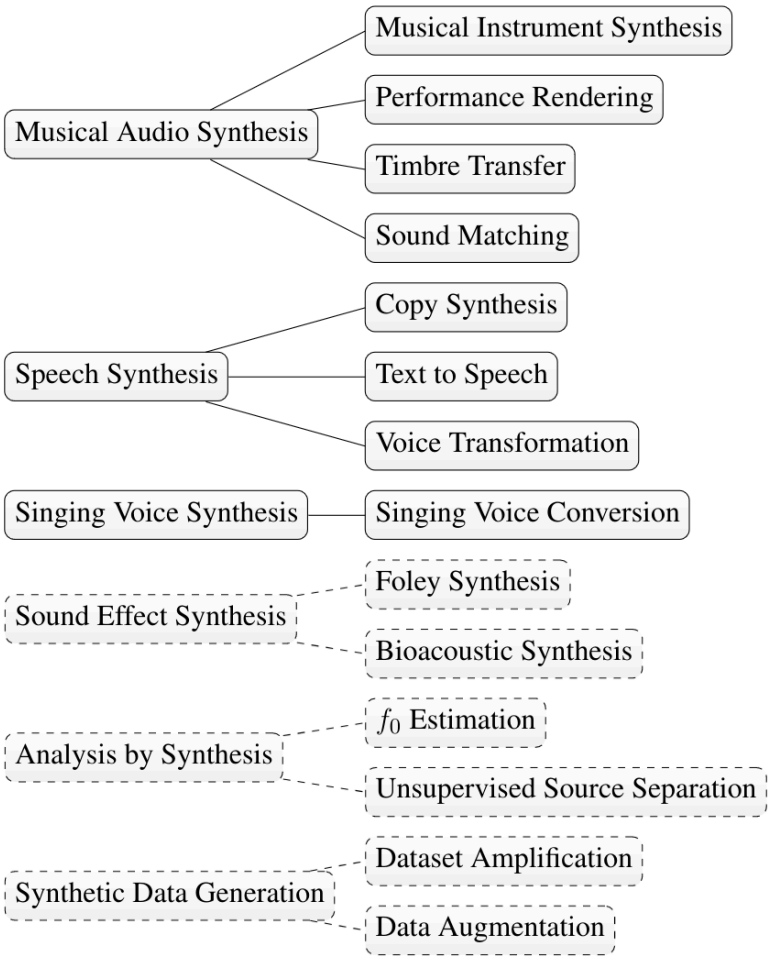


Figure 2. A taxonomy of audio synthesis tasks to which DDSP has been applied. Further discussion on each is presented in section 2. Dashed nodes indicate audio synthesis applications of DDSP found in the literature, but which are beyond the scope of this review.

Amp space limitations

What are the data limitations for this dataset?

https://dafx2020.mdw.ac.at/proceedings/papers/DAFx20in21_paper_47.pdf

The dataset discussed in the document, Amp-Space, is a significant resource for studying fine-grained music timbre transformations, particularly through the lens of musical devices like amplifiers, stompboxes, and studio effects. However, it has several limitations:

> Diversity of Source Waveforms: The dataset uses a small, fixed set of Direct Input (DI) recordings, which might not cover the full scope of possible playing techniques and

frequencies. This could limit the model's accuracy for scenarios significantly different from the training data.
Recording Consistency: Factors such as the use of a single loadbox and interface during dataset creation might not pose issues currently but could require reconsideration in the future to ensure broader applicability.

Generalization: While large neural models and extensive data aim to mitigate these limitations, ensuring that the dataset remains useful even if future versions aim to address these concerns, the inherent complexity of music device timbre makes it challenging to encapsulate entirely within a single resource.

These limitations suggest areas for future improvement and expansion, indicating the evolving nature of research in this domain and the potential need for additional samples to address gaps in the dataset's coverage.

Computation Resources

Apart from **compute Canada cluster**, we can also use the following:

Free Resources

Kaggle: offers a no-setup, customizable, Jupyter Notebooks environment. Here, you will have access to Nvidia P100 GPUs for ~30-38h per week (changes weekly), and an 8-core TPUv3 for 20h per week. You can run notebooks interactively and also in the background for up to 8h (when clicking on "Save and Commit"). You have access to up to 100GB of private storage (known as "Datasets") and unlimited public storage.

TPU Research Cloud: TRC enables researchers to apply for access to a cluster of more than 1,000 Cloud TPUs. Researchers accepted into the TRC program will have access to v2 and v3 devices at no charge and can leverage a variety of frameworks including TensorFlow, PyTorch, Julia and JAX. NOTE: Please apply as soon as possible as it takes time for them to approve.

Paid Options

Azure: When you sign up to Azure, you can get $200 USD in free credits. You may need a credit card (you can buy a prepaid visa card in a grocery store) and/or a phone number. Once you signed up, you can run experiments in a Jupyter Notebook.

Diagrams saved for future reference:

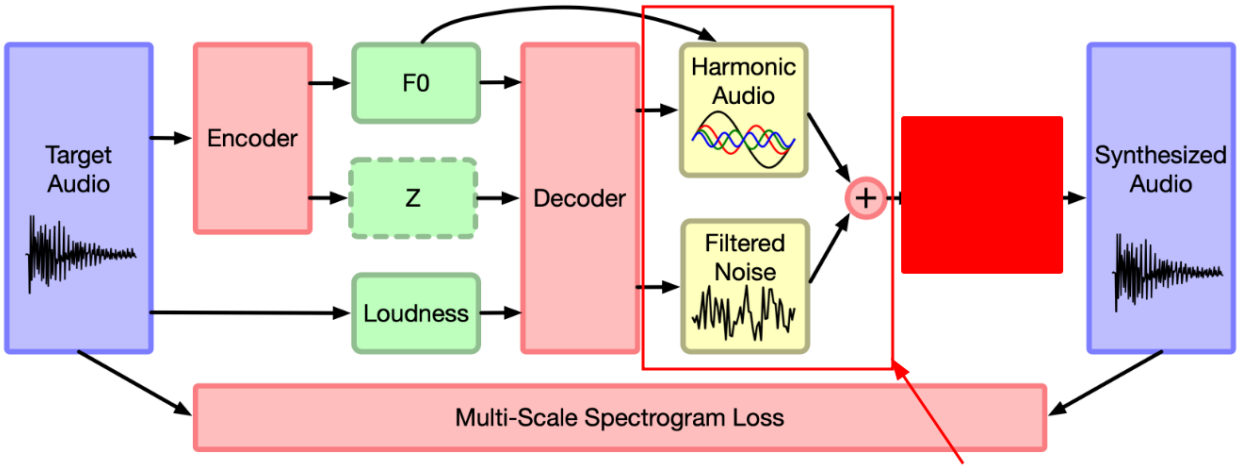


Figure 2: Autoencoder architecture. Red components are part of the neural network architecture, green components are the latent representation, and yellow components are deterministic synthesizers and effects. Components with dashed borders are not used in all of our experiments. Namely, z is not used in the model trained on solo violin, and reverb is not used in the models trained on NSynth. See the appendix for more detailed diagrams of the neural network components.

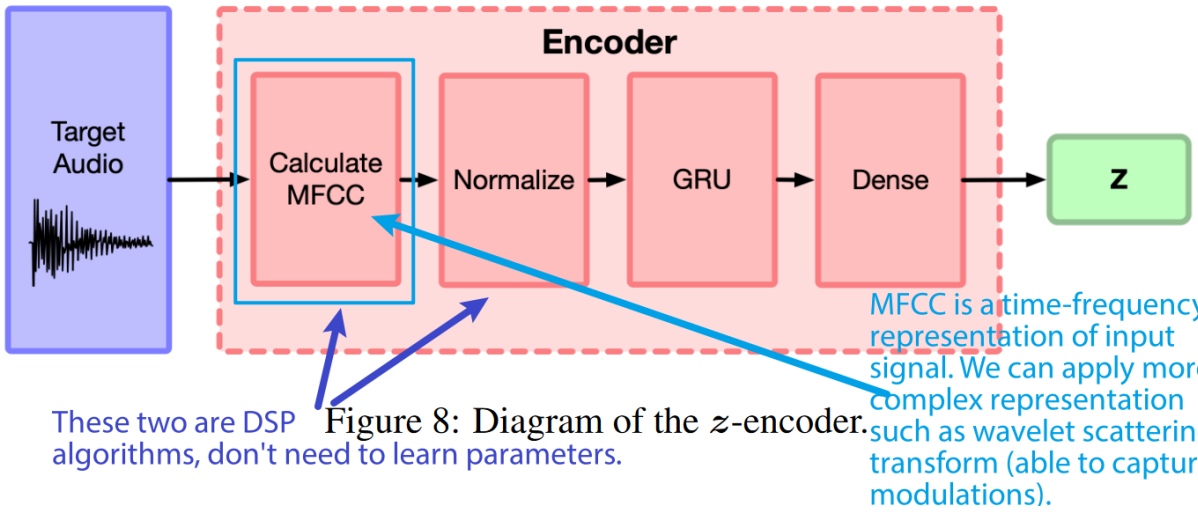


Figure 8: Diagram of the z-encoder.

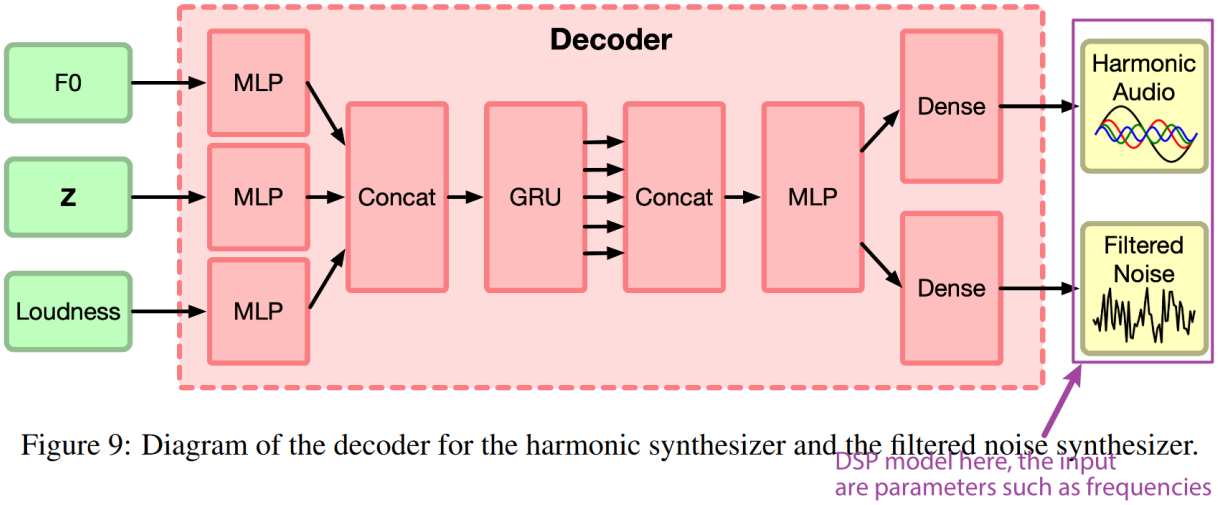


Figure 9: Diagram of the decoder for the harmonic synthesizer and the filtered noise synthesizer.