Music generation with Diffusion models - updates

Bojescu Mihai, 2025

Agenda

- Progress
- Future work

Progress

Progress

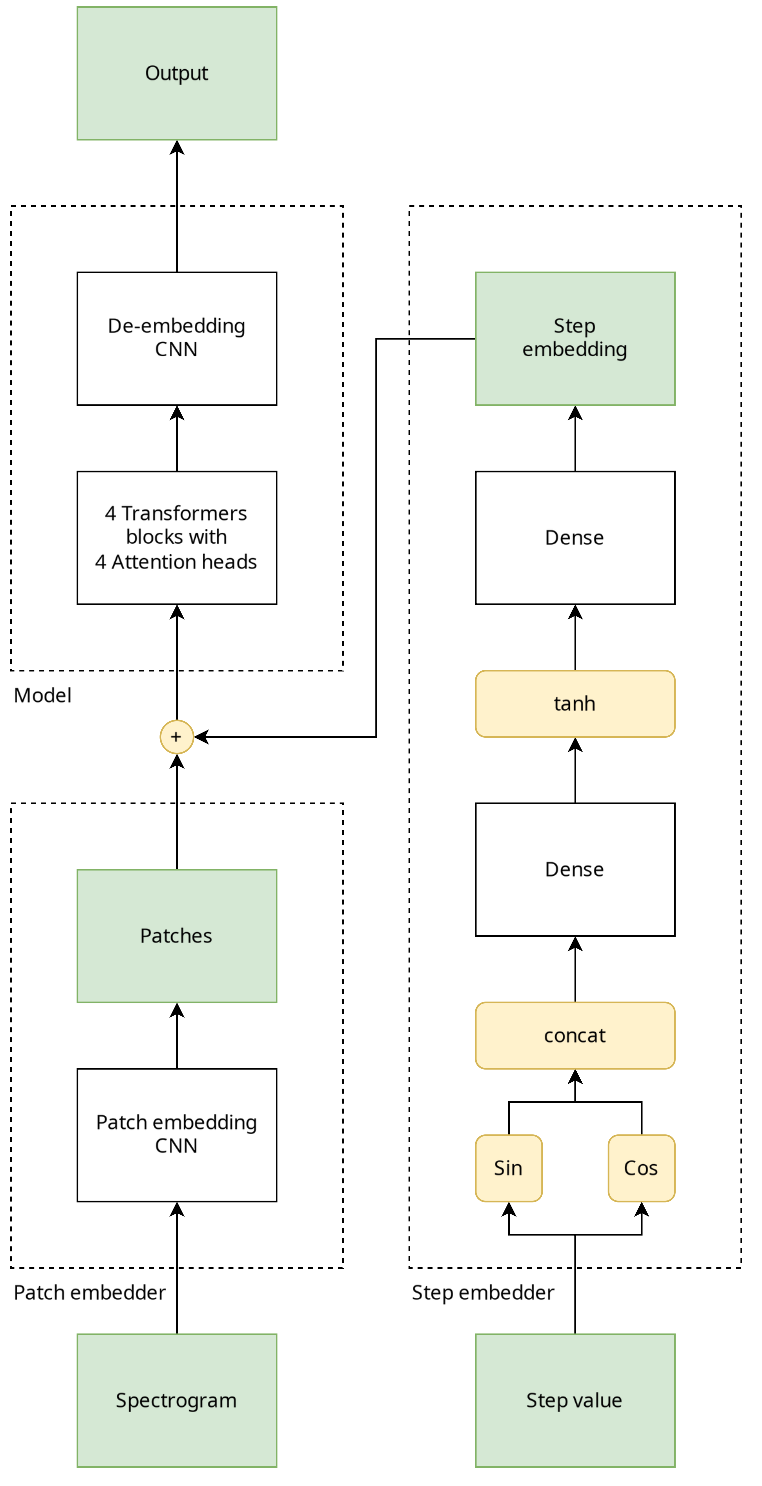
- Data
- Model
- Pipeline:
  - Preprocessing
  - Training loop

Data

- The dataset is composed of:
  - 710 songs, with different genres, some with lyrics
  - 1106 labels, with multiple labels per song
- The dataset is freely available on *Huggingface*

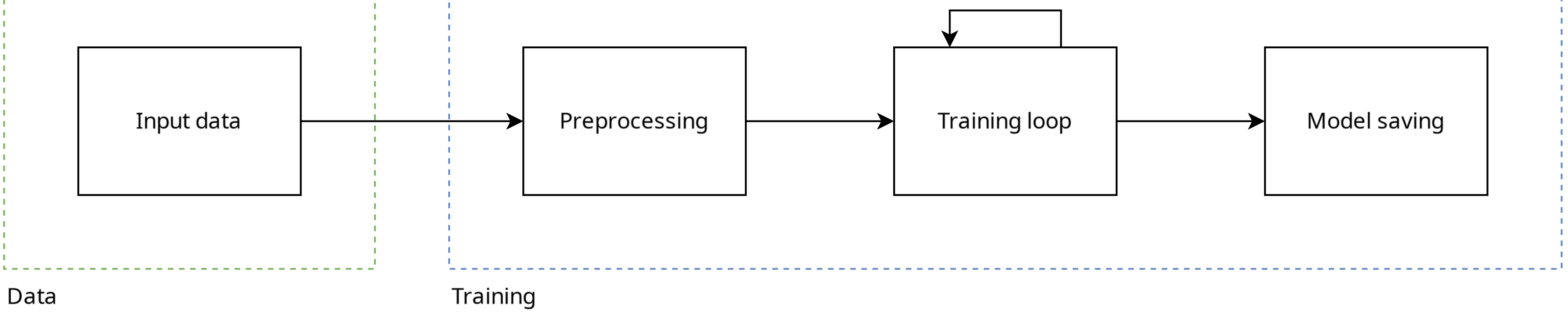
Model

- Currently, the model is a "*Diffusion Transformer*", with:
  - 4 Transformer blocks, with 4 Attention heads each
  - Patch embedding and de-embedding is done using CNNs
  - Timestep embedding and is done via a Dense network
- Number of parameters is 5 388 802



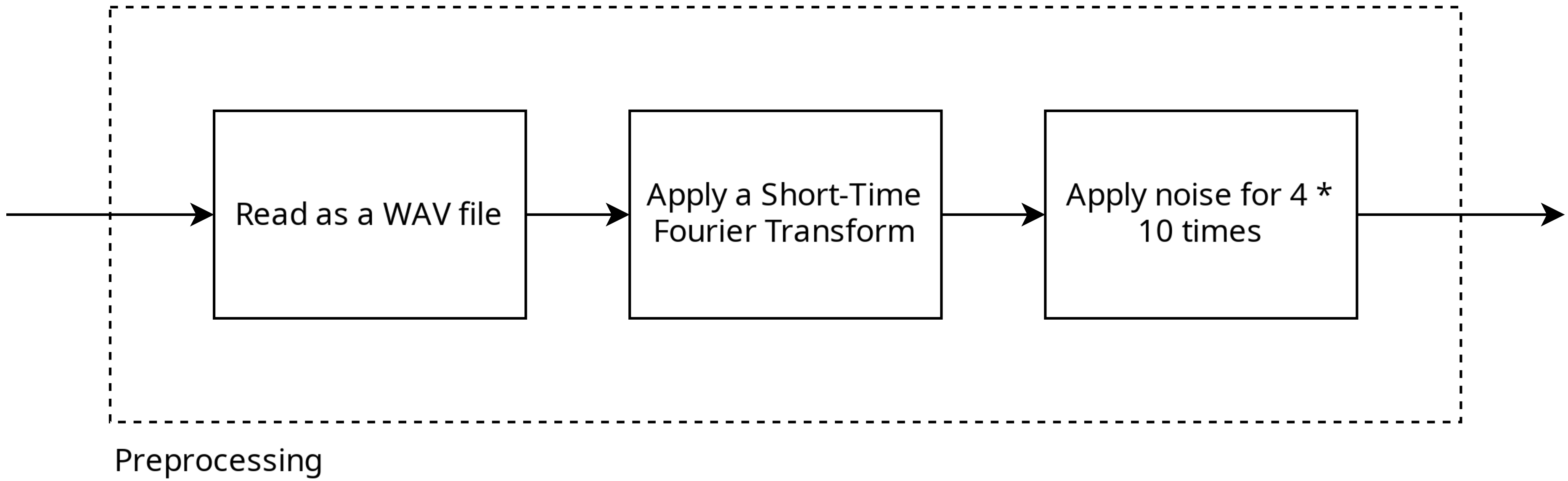
Output
De-embedding CNN
4 Transformers blocks with 4 Attention heads
Model
+
Patches
Patch embedding CNN
Patch embedder
Spectrogram
Step embedding
Dense
tanh
Dense
concat
Sin
Cos
Step embedder
Step value

Pipeline



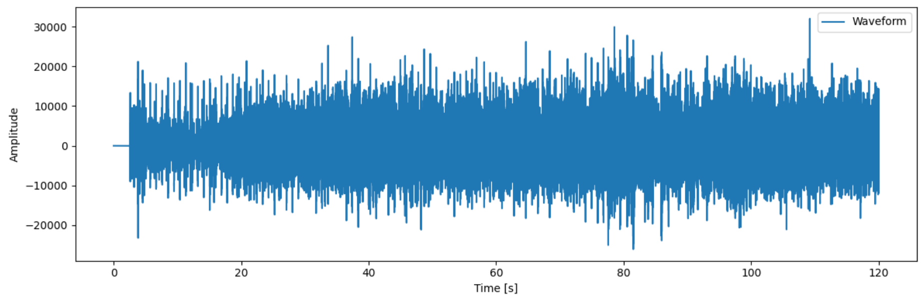
Input data
Preprocessing
Training loop
Model saving
Data
Training

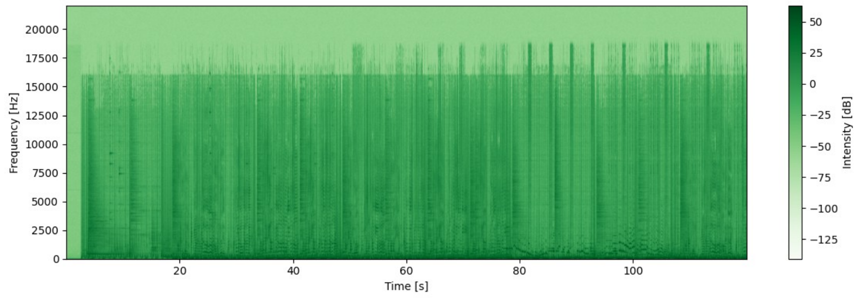
Preprocessing



Preprocessing

- Given a waveform, a Short-Time Fourier Transform is applied
- A spectrogram is given as a result
- The spectrogram is saved as a tensor

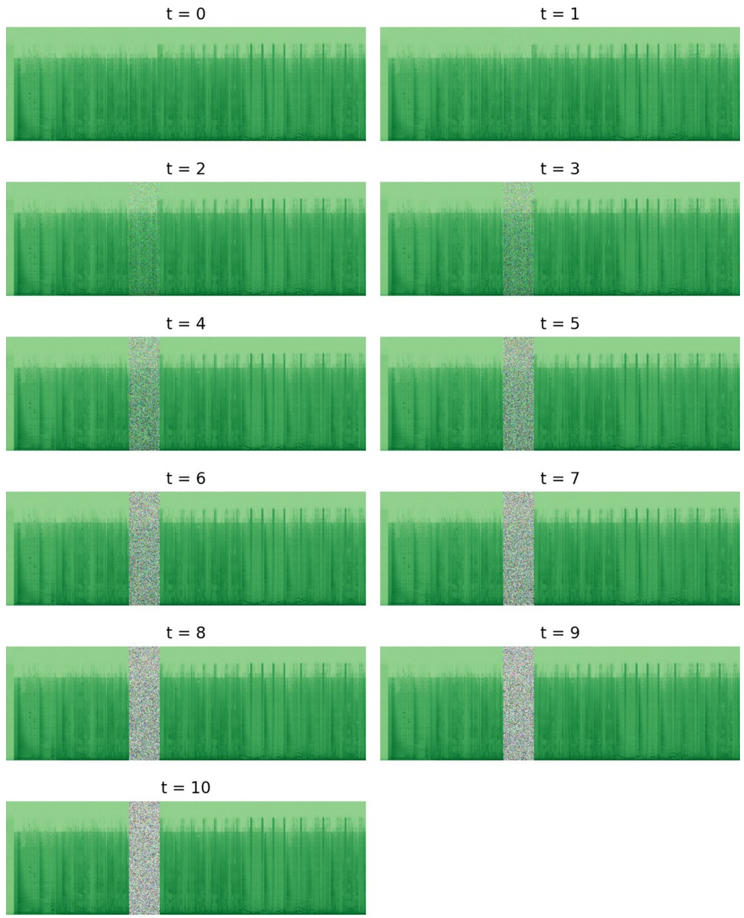




Preprocessing

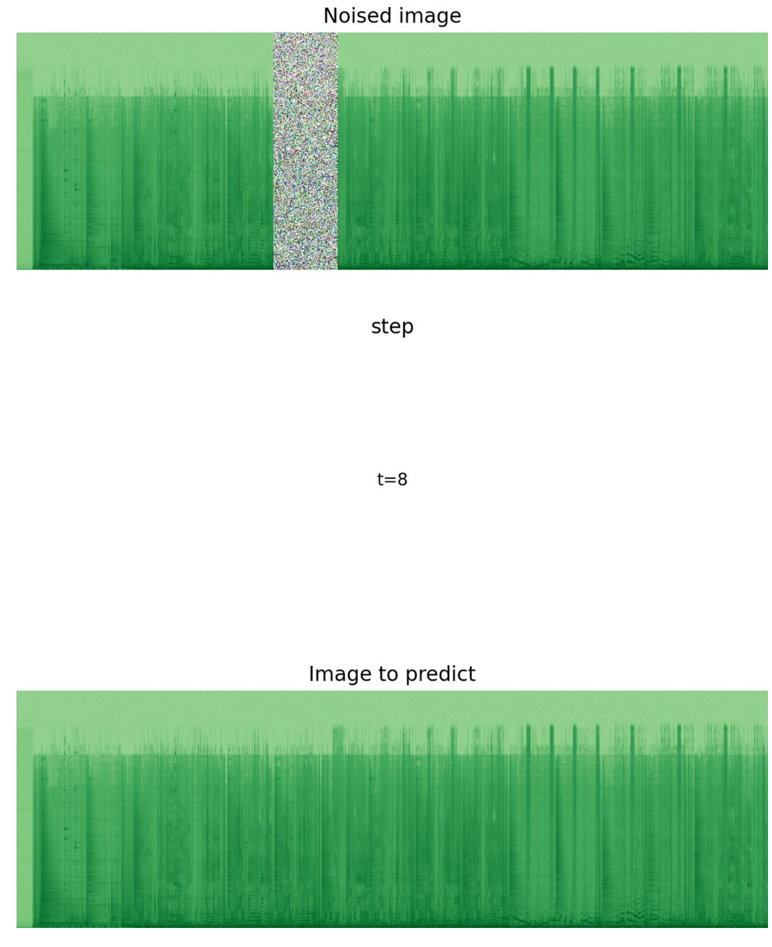
- The spectrogram is gradually "noised":
  - Start with an unnoised spectrogram
  - For *t* iterations, add more noise
- To get a better result, the noise scheduling process described in the "*Denoising Diffusion Probabilistic Models*" (Ho et al.) paper is used
- The formula is:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$$



Training loop

- At training, the model is given:
  - A noised spectrogram
  - A timestep value
- The model is tasked to predict the original image
  - **Motivation**: the model does not need to capture intermediary denoising steps, as it introduces noise
- The loss is influenced by the difference between:
  - The predicted denoised spectrogram
  - The actual spectrogram



Training loop

- Training loss: MSE
- Training optimizer: Adam
- Training statistics:
  - Uses about 8GB of VRAM for training
  - Training takes 5 epochs * 450s ~= 2250s on 1 GPU

Future work

Future work

- Add:
  - support for text-driven generation
- Explore with:
  - different input tensor shapes and sizes
  - different learning parameters
  - different model sizes
  - different model configurations
- Training:
  - run training multiple times

Thank you!