# Generating Complementary Instrument
# Tracks with a Transformer Model

Mikko Murhu

# Introduction

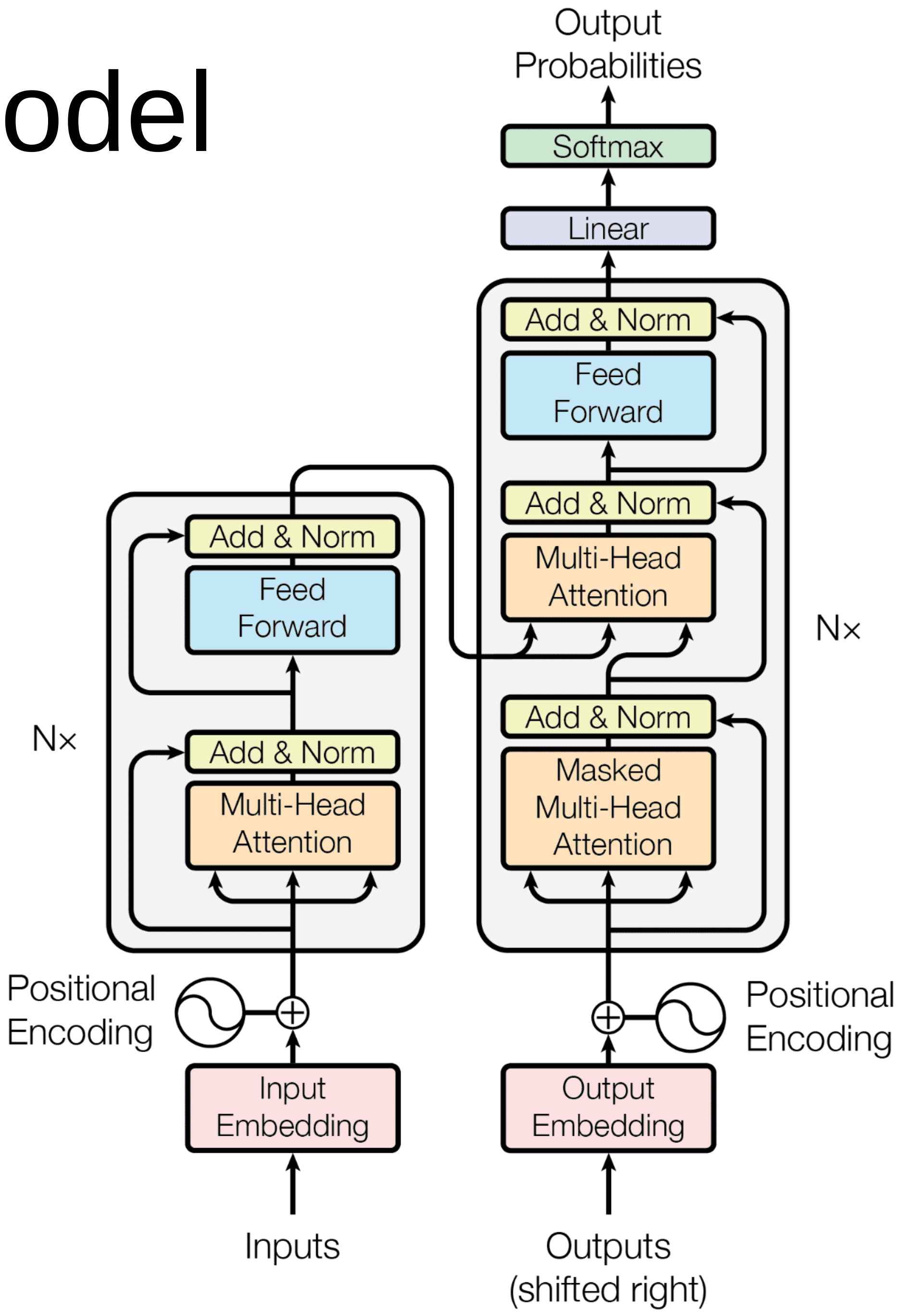
The objective: explore the capabilities of an encoder-decoder transformer model in generating complementary instrument tracks to existing instrument tracks

# Introduction – why transformers?

- Transformers are used in natural language processing for many purposes, including translation
- Translation converts instrument tracks to a set of accompanying instrument tracks
- Previously decoder-only transformers have been used for less conditioned music generation

# Transformer model

- A deep neural network
- Processes sequential data efficiently
- Replaces recurrency with attention mechanism and positional encoding



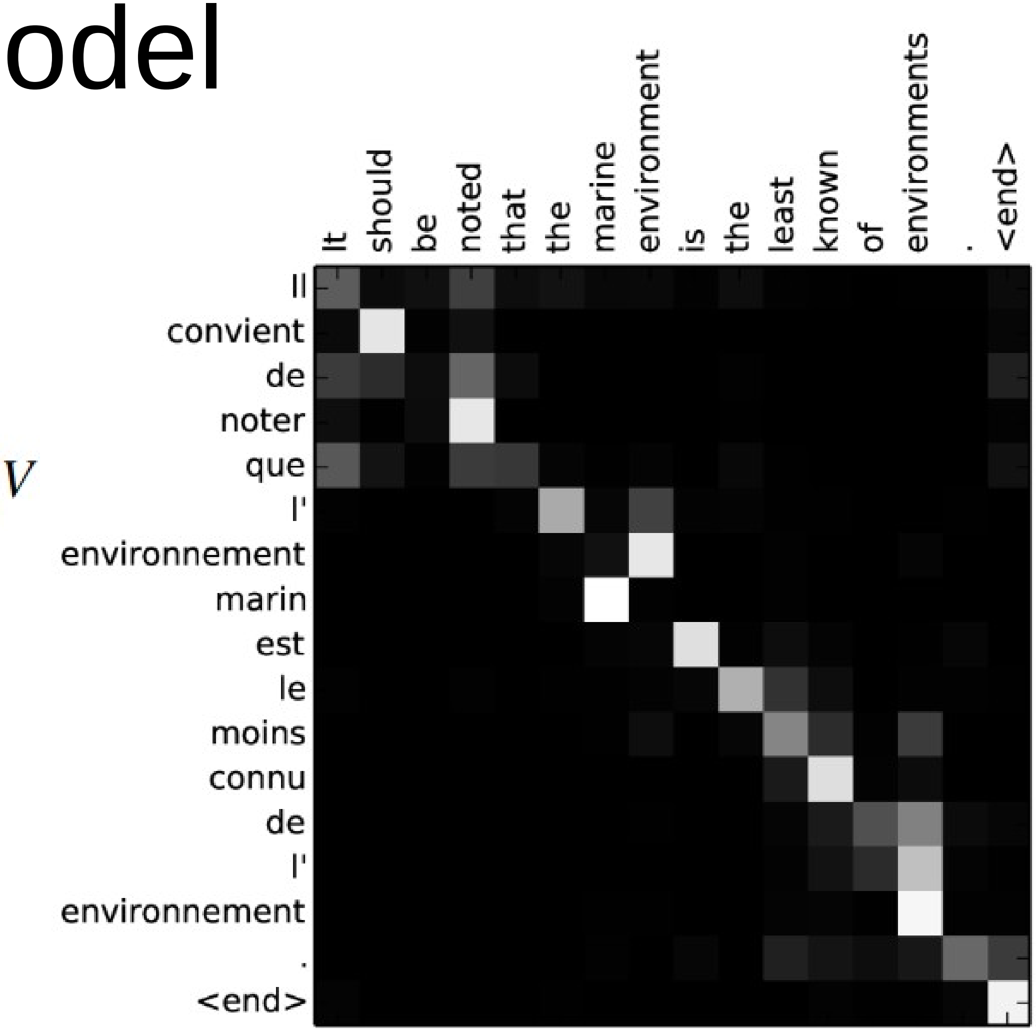
# Transformer model

### Attention mechanism

- Defined by

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^{\top}}{\sqrt{d_k}})V$$

- Calculates strengths of relationships of elements in sequences
- Three different contexts in an encoder-decoder transformer:
  - Self-attention in encoder
  - Self-attention in decoder
  - Cross-attention in decoder



An attention score heatmap

# Transformer model

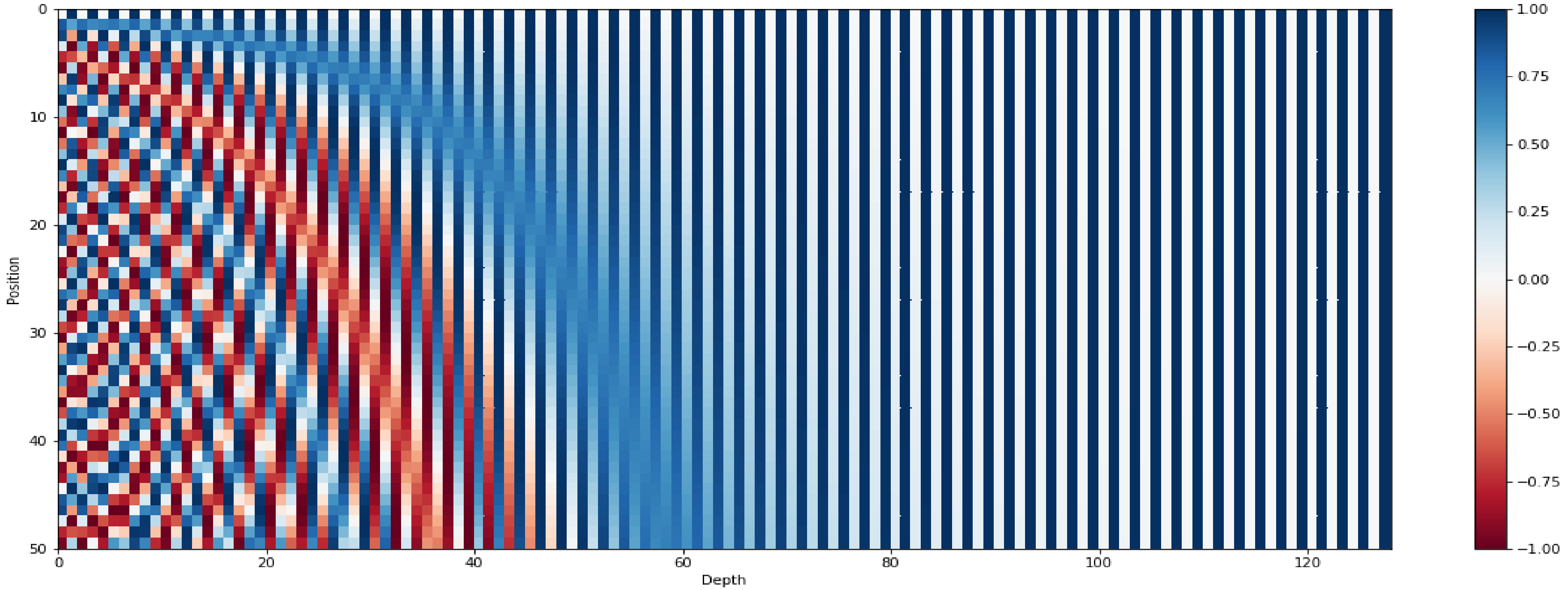
## Positional encoding

- A unique vector for each element in sequence

$$PE(pos, 2i) = \sin(pos/n^{(2i/d)})$$

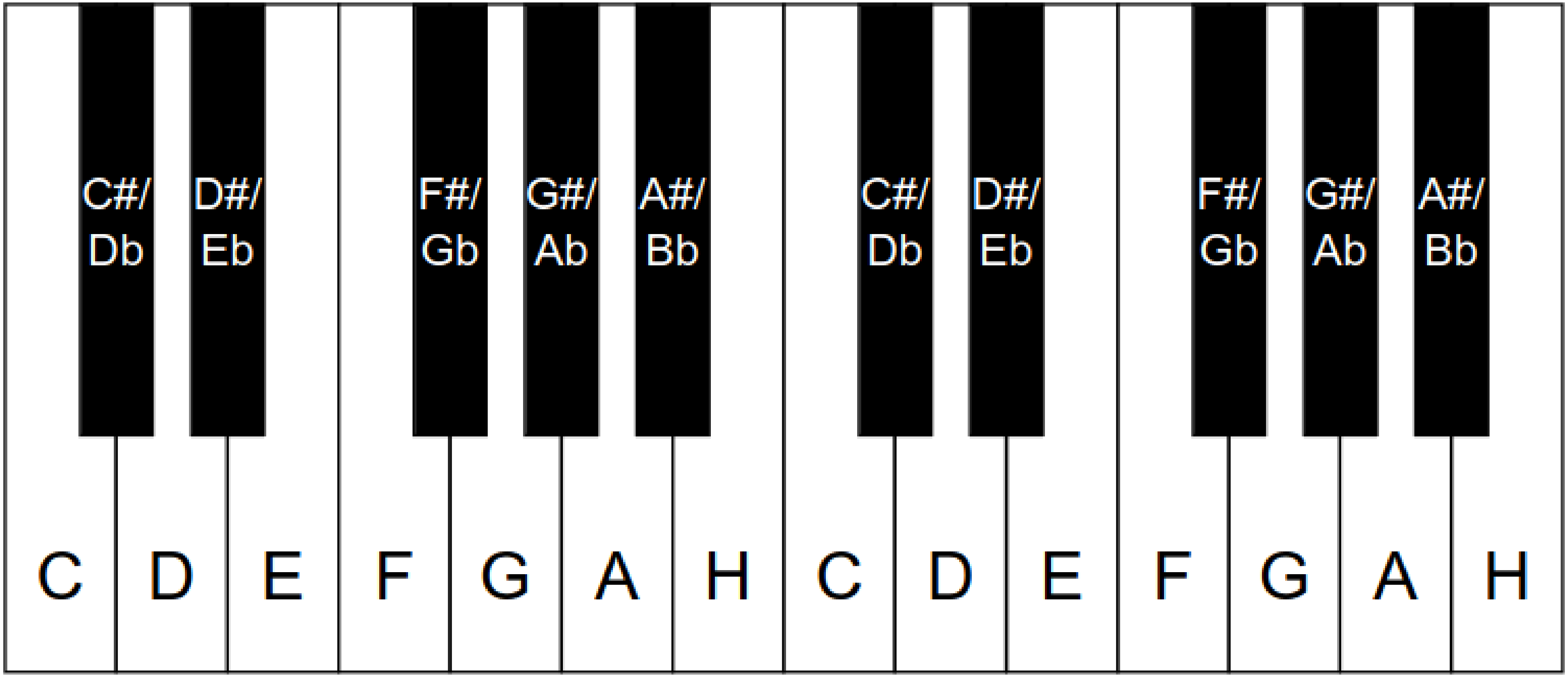
$$PE(pos, 2i+1) = \cos(pos/n^{(2i/d)})$$

- Added to embedding vectors
- Injects positional information to the embedding



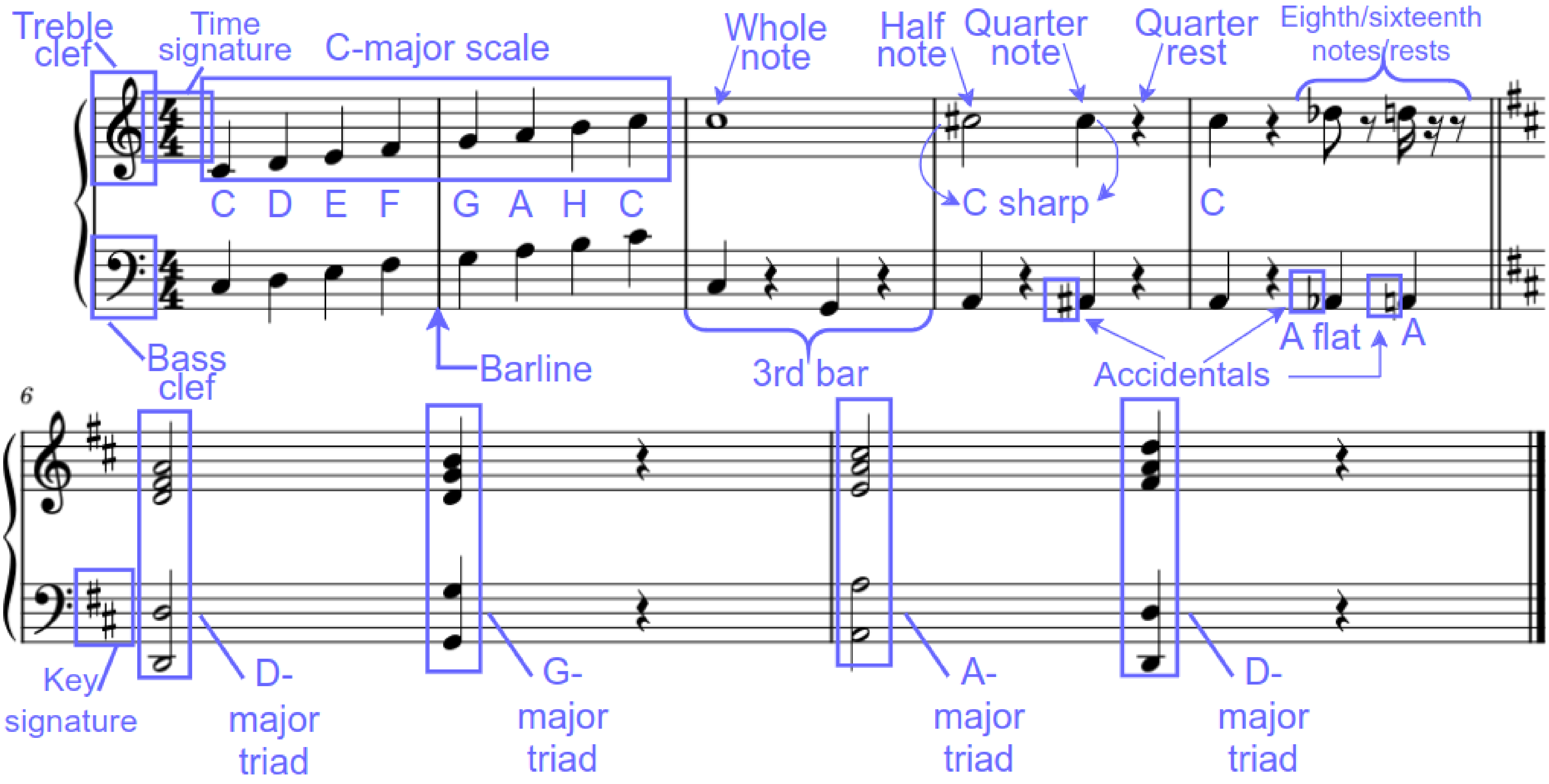
**Positional encodings from positions 0 to 50 with model dimension 128.**

# Elements of music



- 7 named pitches, including raised/lowered pitches create 12 unique pitches
- Harmony: key, scales, chords
- Rhythm: pulse, beat

# Elements of music

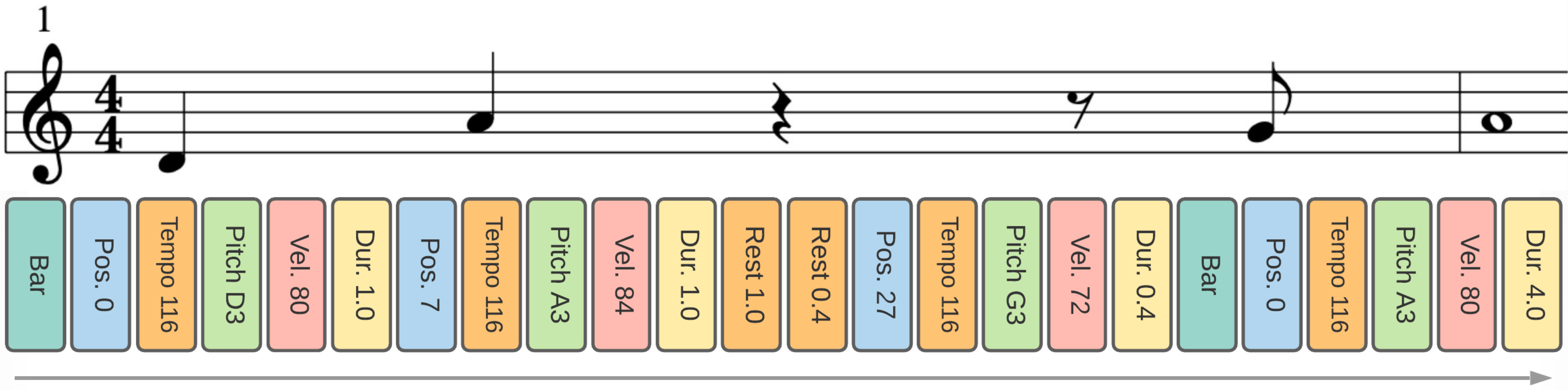


Music notation – sheet music

# Dataset

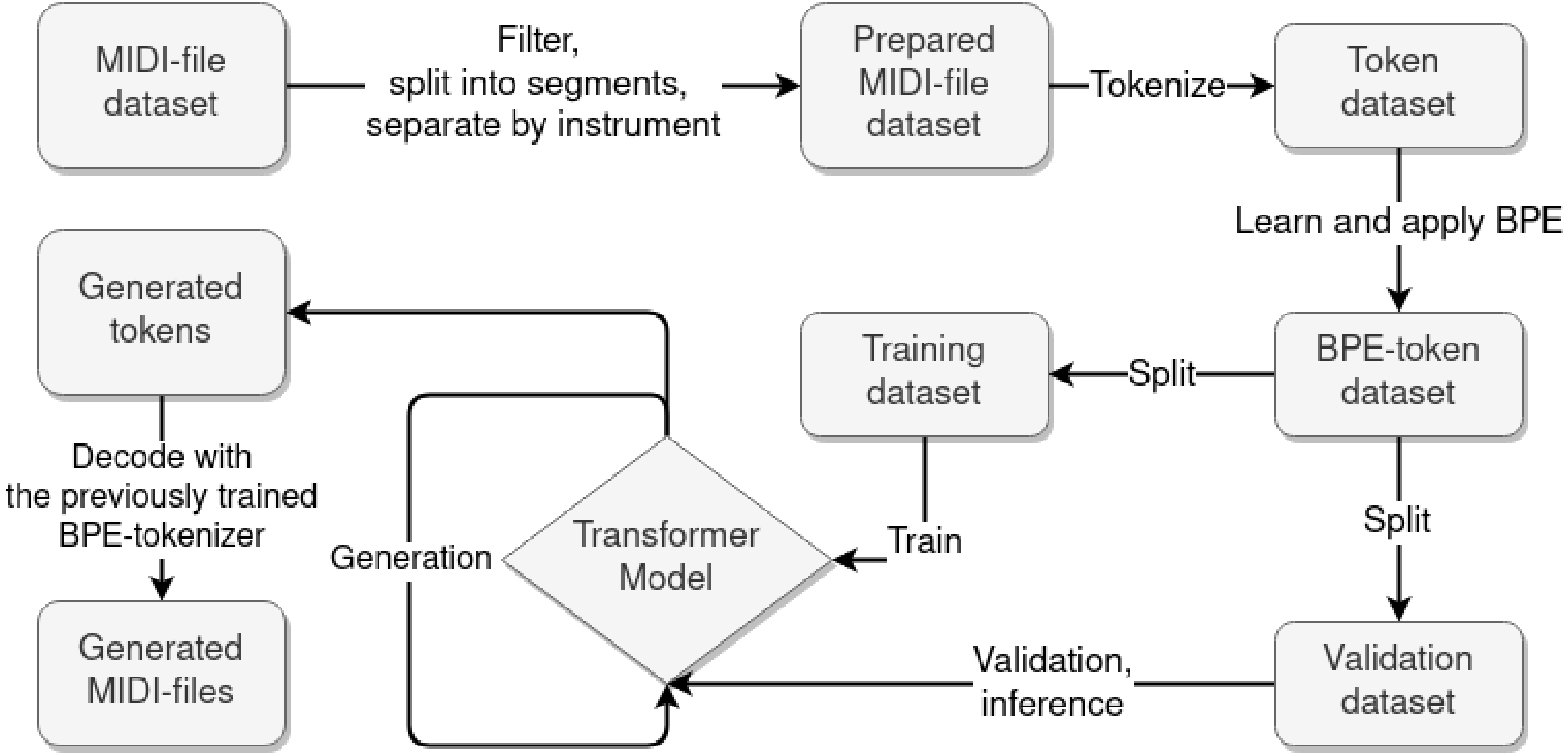
- MIDI files characterize music with discrete time-based events
- Lakh MIDI dataset has 176581 unique midi-files
- The thesis uses the "clean subset" of the Lakh dataset, which has 17233 midi-files

# Tokenization



- REMI-tokenization scheme
- Byte pair encoding (BPE)

# Steps in the pipeline



# Quantitative results

**Setup**

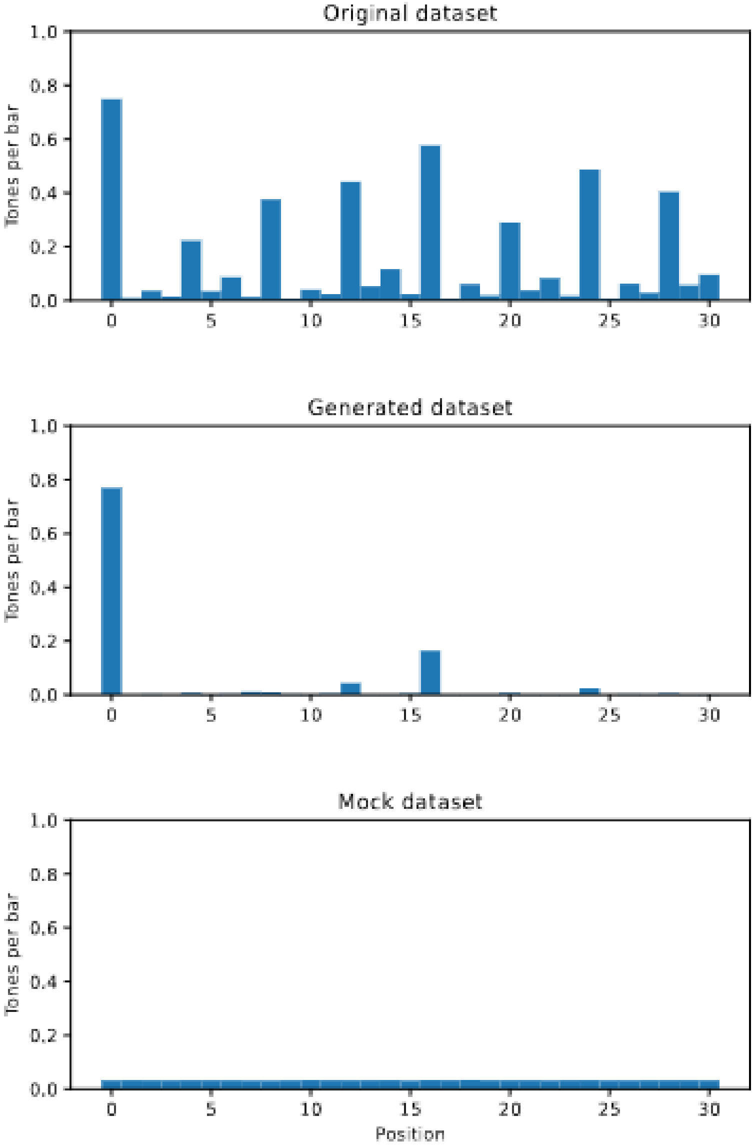
- 300 attempted generated samples, which resulted in 299 piano and 294 bass samples
- Rhythm analysis
- Harmony analysis

# Quantitative results

## Rhythm analysis

- Distributions of notes per bar
- Separate bars into discrete rhythm representation strings ("101...01")
- Sample Levenshtein distances with string pairs between generated sample vs. validation sample
- Compare with mock sample vs. validation sample

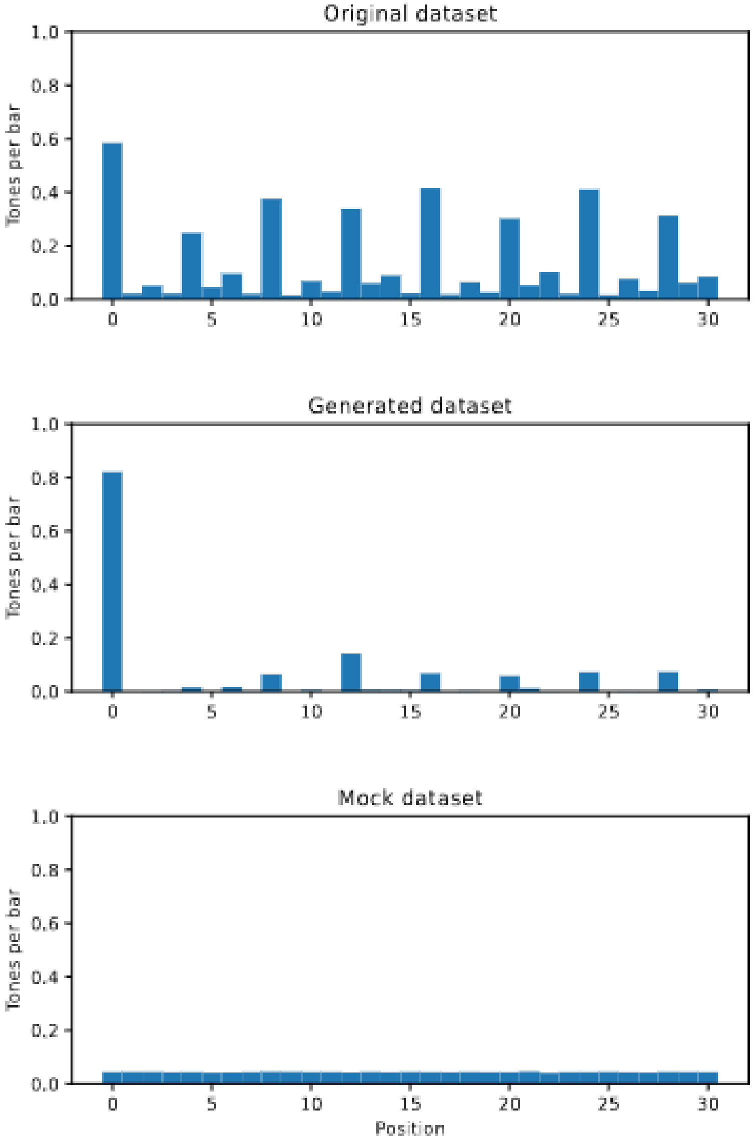
# Quantitative results



Bass note distributions

|  | Bass val. | Piano val. |
|---|---|---|
| Gen | 4.104 | 4.103 |
| Mock | 4.973 | 4.800 |

**Avg. Levenshtein distances**



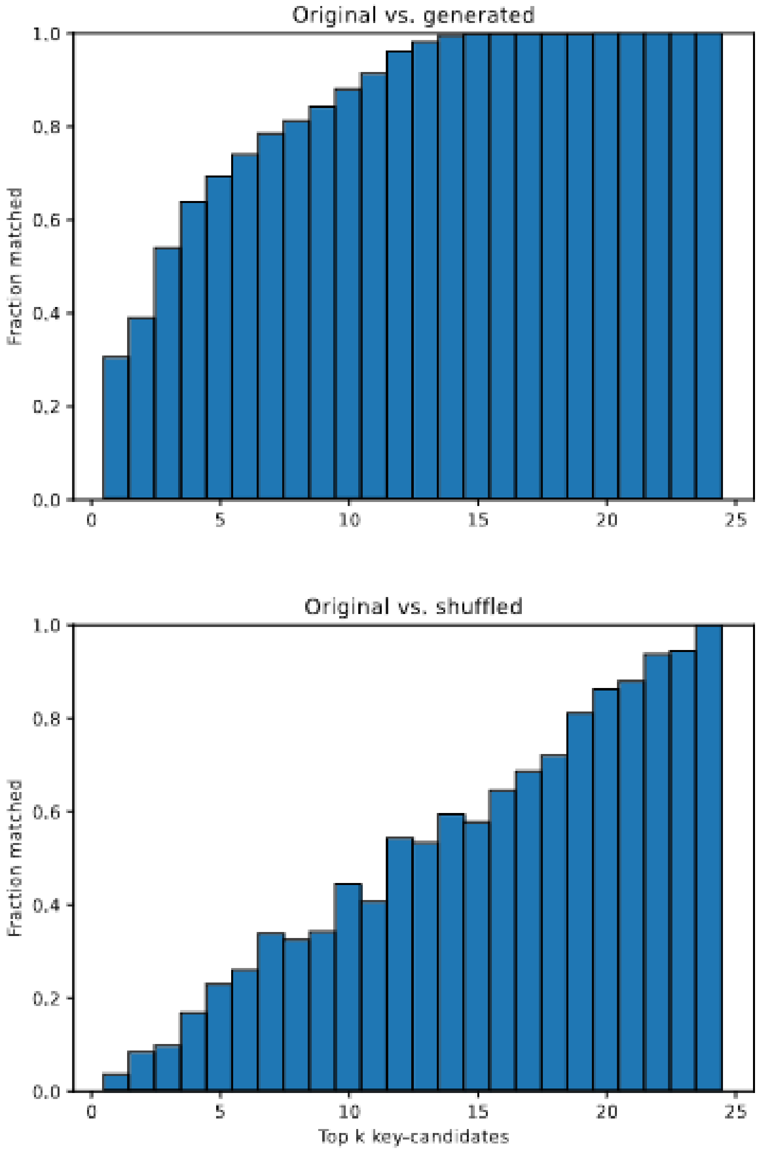
Piano note distributions

# Quantitative results

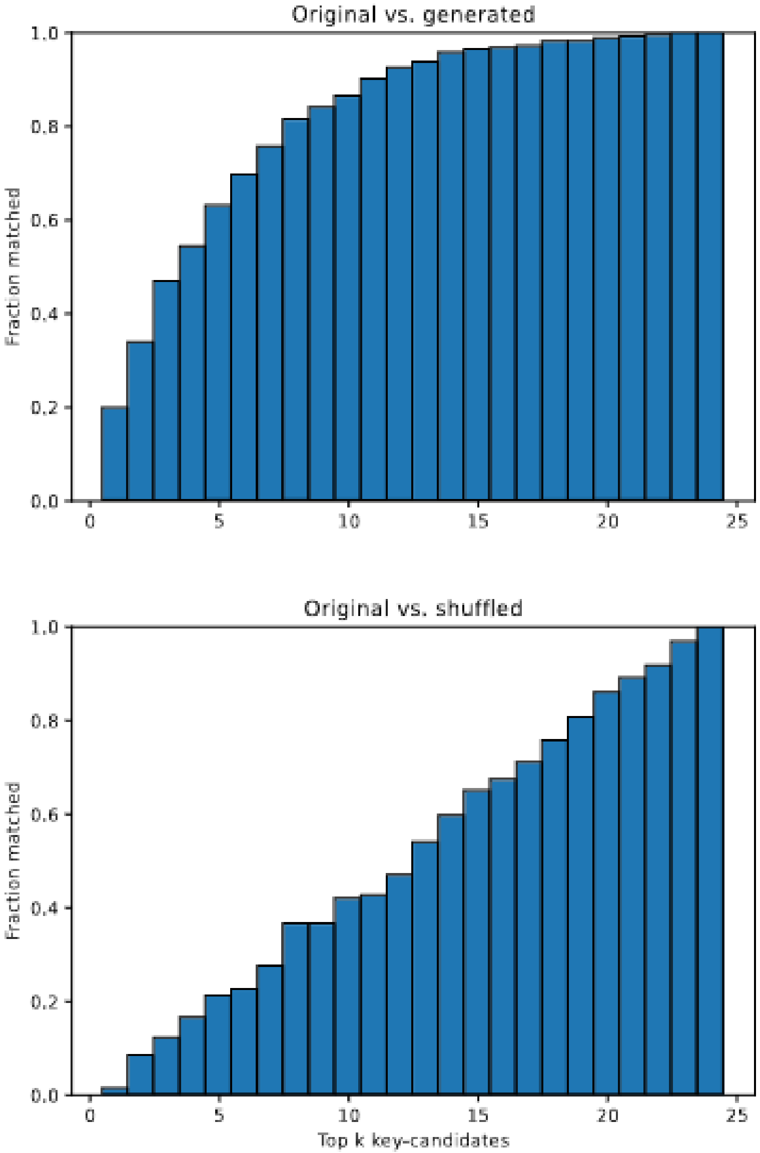
**Harmony analysis**

- Determine what *key* the samples are in with the Krumhansl-Schmuckler key-finding algorithm
- Compare the top-k key candidates given by the algorithm with the key of the reference

# Quantitative results

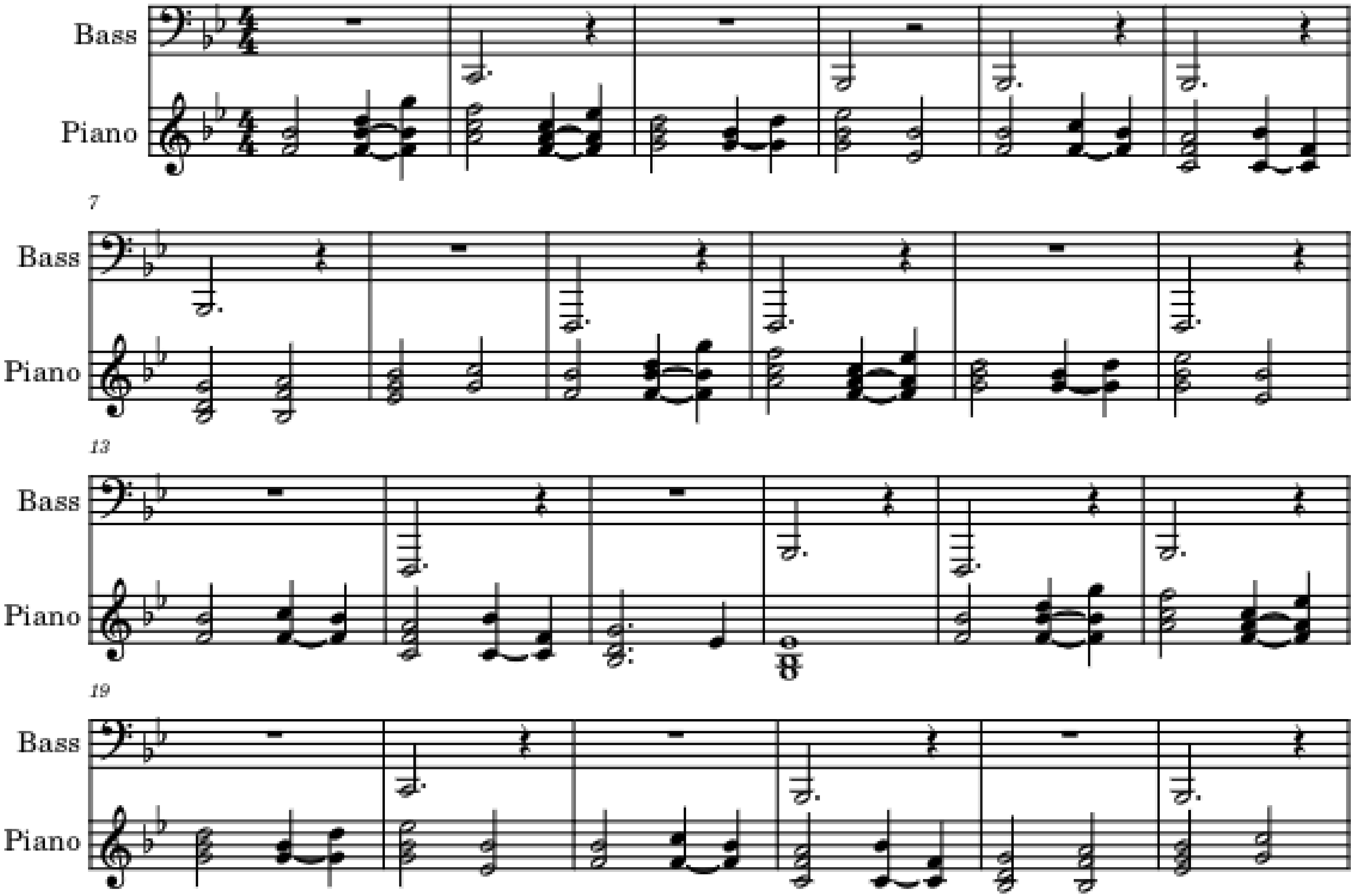


**Bass key comparisons**



**Piano key comparisons**

# Qualitative results



# Discussion

- Model performance is unimpressive
- The model learns some larger-scale information about the reference piece but does not react to smaller-scale changes
- Possible reasons:
  - Small/bad-quality dataset
  - Training parameters
  - Tokenization/model parameters
  - Inference method
- Model could potentially perform better if these reasons are addressed