

AI Music

Ivan Shanin, PhD student @ Centre for Digital Music, Queen Mary University of London

ivan.shanin@qmul.ac.uk

Part 1. Intro

What is music data?

How can we apply AI to Music?

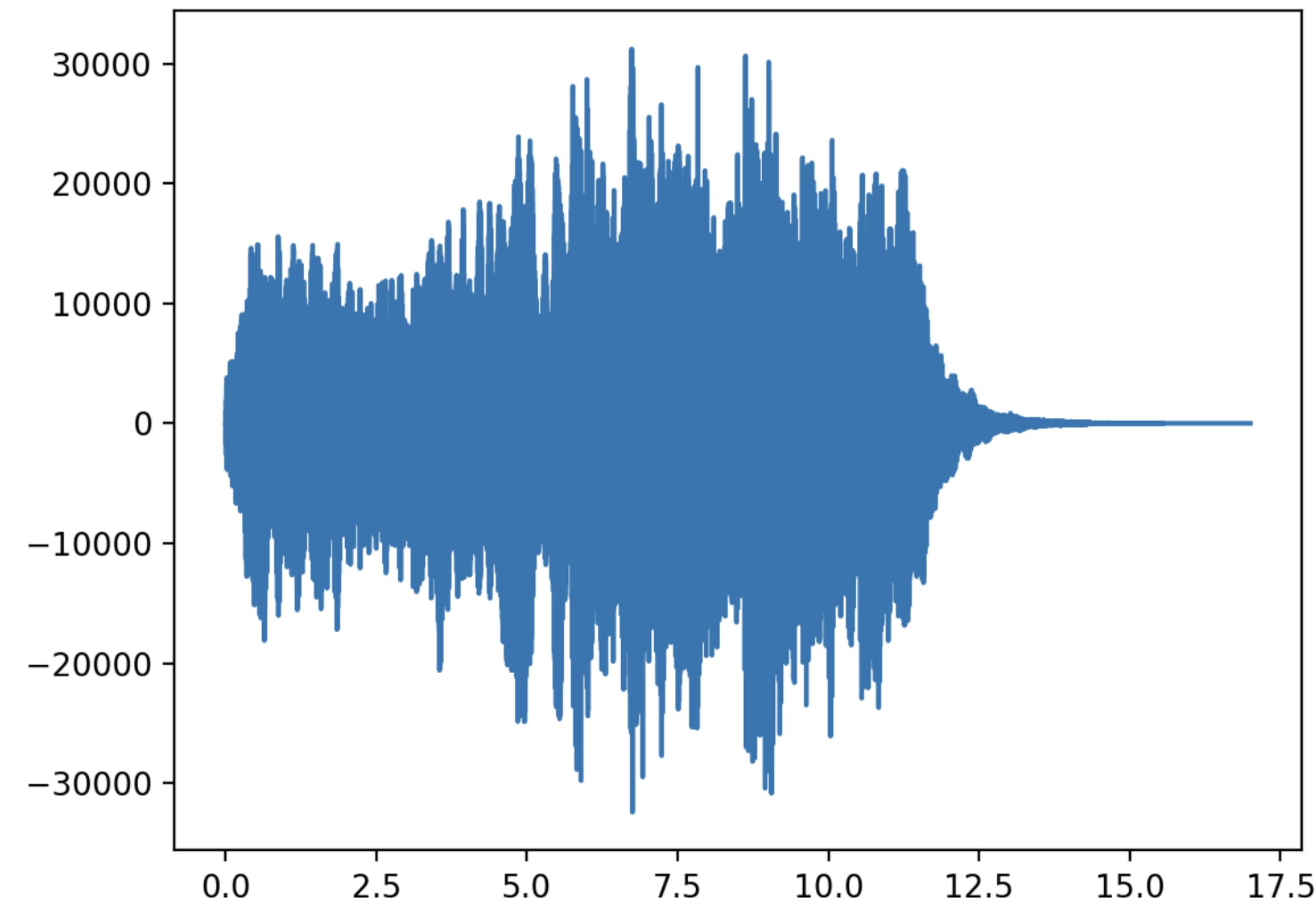
Music Data

Waveform

high-dimensional (e.g. 44.1 kHz)

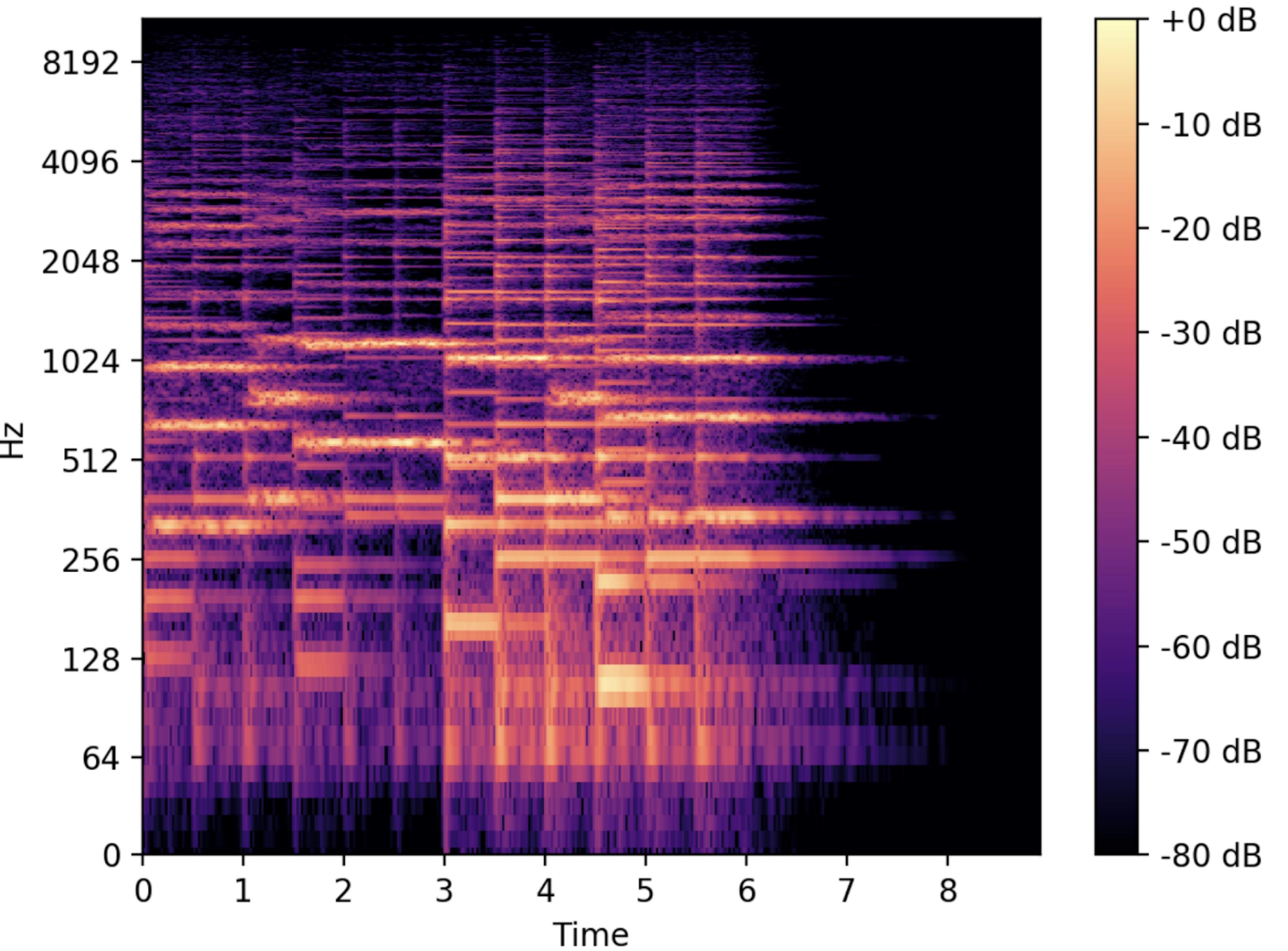
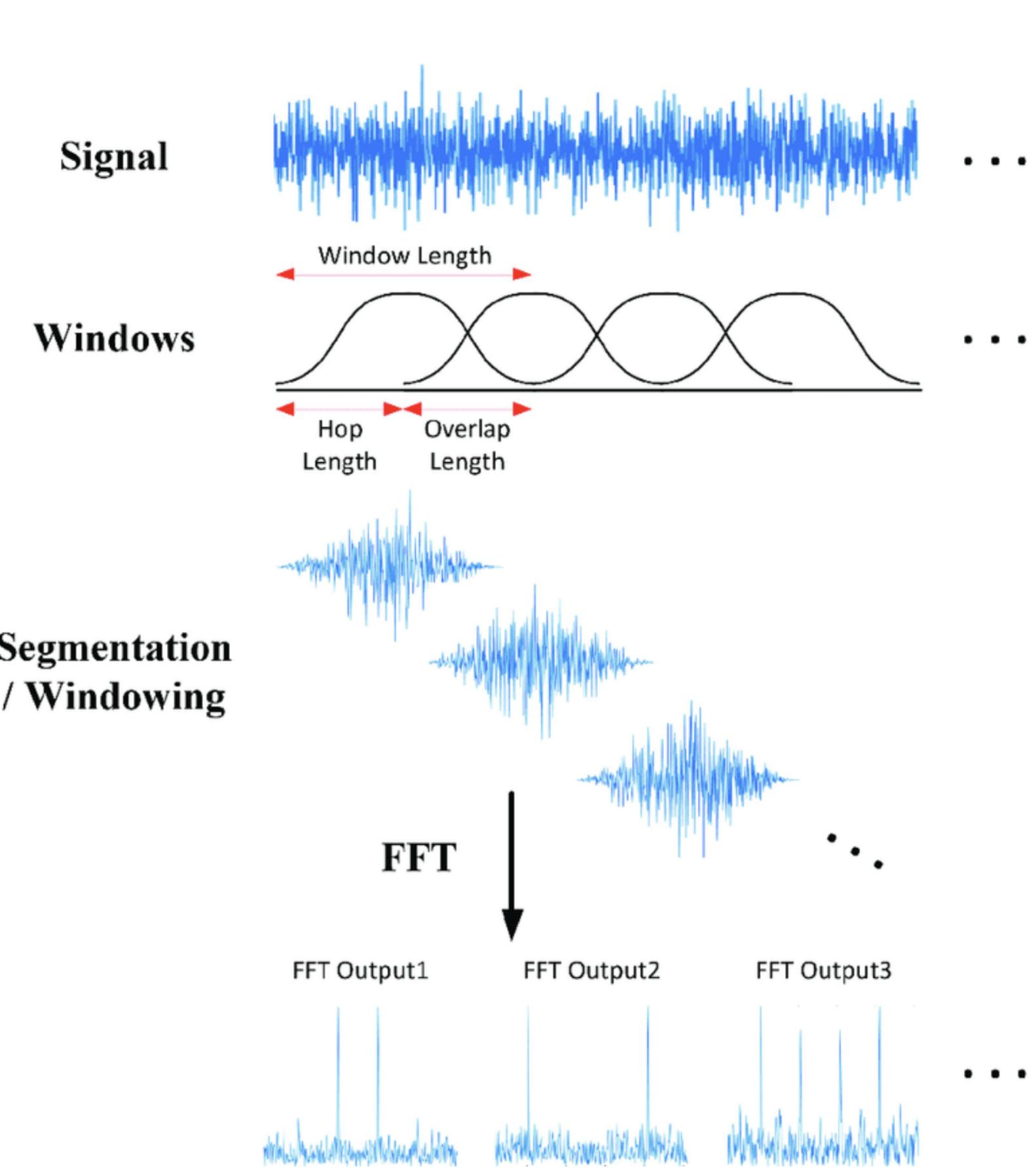
visually uninterpretable

the most “raw” representation



Music Data

Short-time Fourier Transform



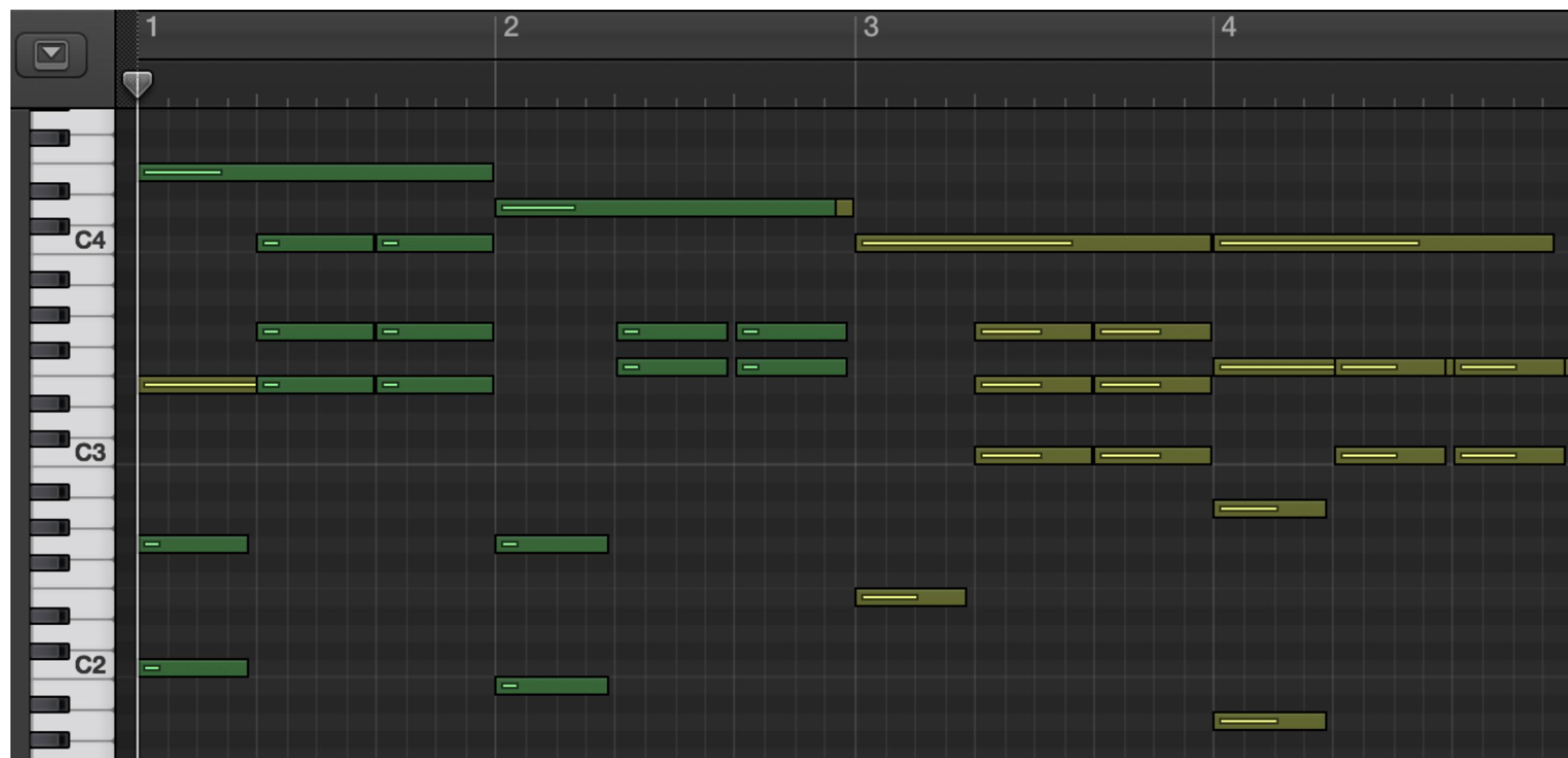
Music Data

MIDI: Music Instrument Digital Interface

symbolic representation

pitch, onset, timbre information

very compact, widely used in
professional community and industry



Music Data

Musical Scores



Figure 1.14 from [Müller, FMP, Springer 2015]

Music Data

MusicXML

```
<note>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
</note>
```



Figure 1.15 from [Müller, FMP, Springer 2015]

AI Music Research Scope

Music Information Retrieval:

music tagging, classification, emotion recognition, music transcription, extracting rhythmic and harmonic information, music structure analysis, music search, acoustic fingerprinting

AI Music Research Scope

Music Information Retrieval:

music tagging, classification, emotion recognition, music transcription, extracting rhythmic and harmonic information, music structure analysis, music search, acoustic fingerprinting

Musical Audio Signal Processing:

music source separation, digital audio effects, differentiable digital signal processing, timbre transfer and analysis, neural audio synthesis

AI Music Research Scope

Music Information Retrieval:

music tagging, classification, emotion recognition, music transcription, extracting rhythmic and harmonic information, music structure analysis, music search, acoustic fingerprinting

Musical Audio Signal Processing:

music source separation, digital audio effects, differentiable digital signal processing, timbre transfer and analysis, neural audio synthesis

Music Generation:

text-to-music, music editing, music inpainting, human-computer co-creation

AI Music Research Scope

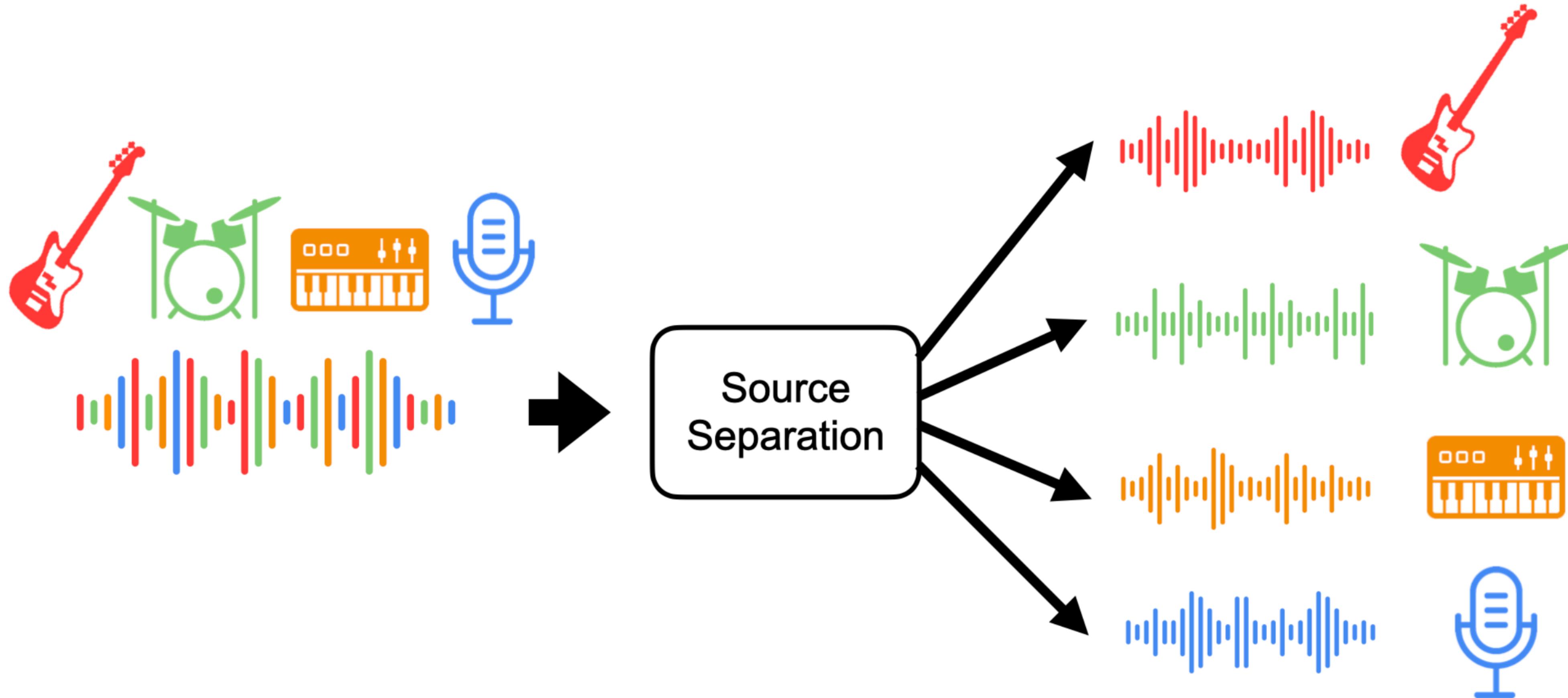
Interdisciplinary topics:

Music Cognition / Perception

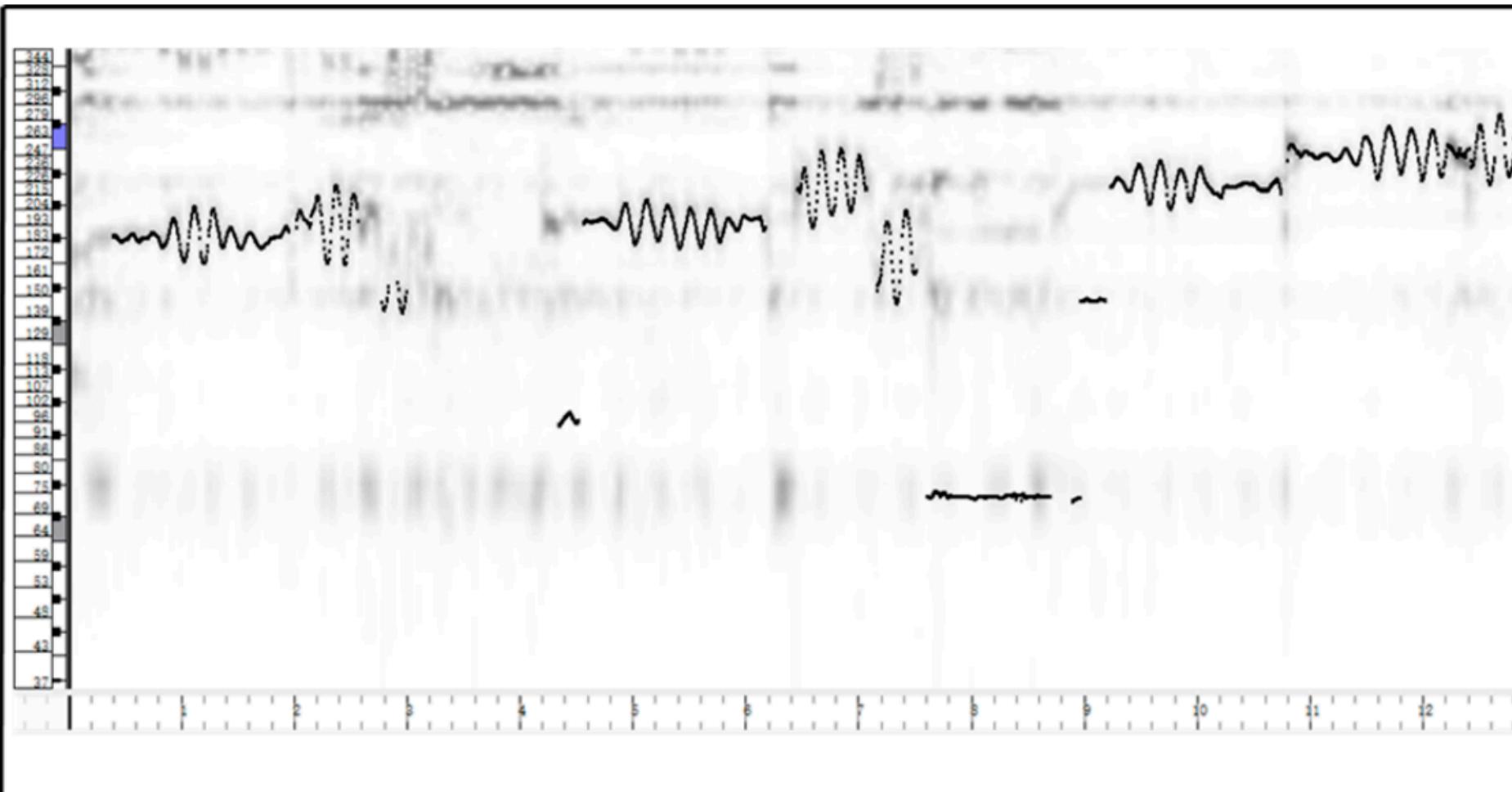
Music and Health

Optical Music Recognition

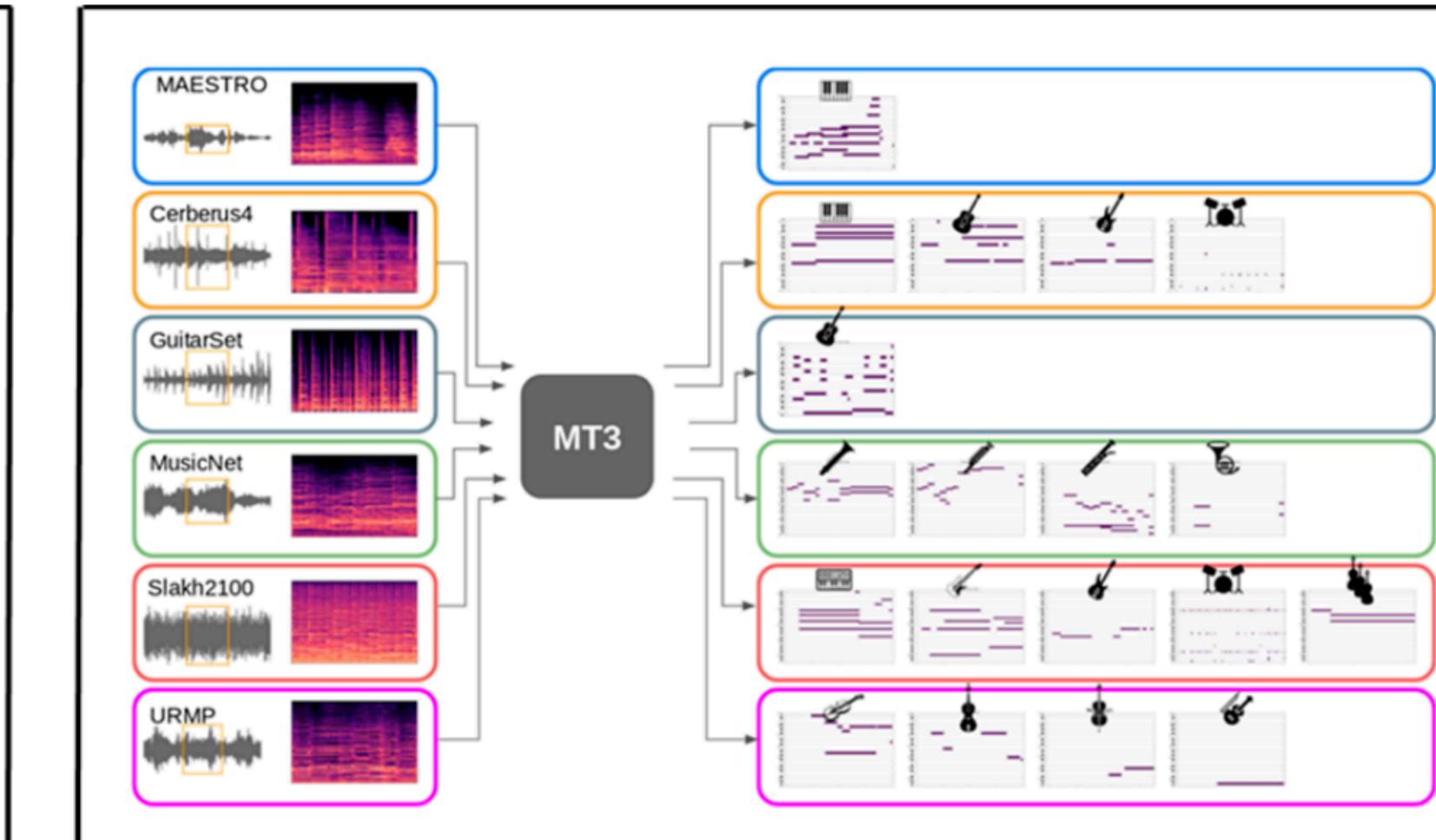
Music Source Separation



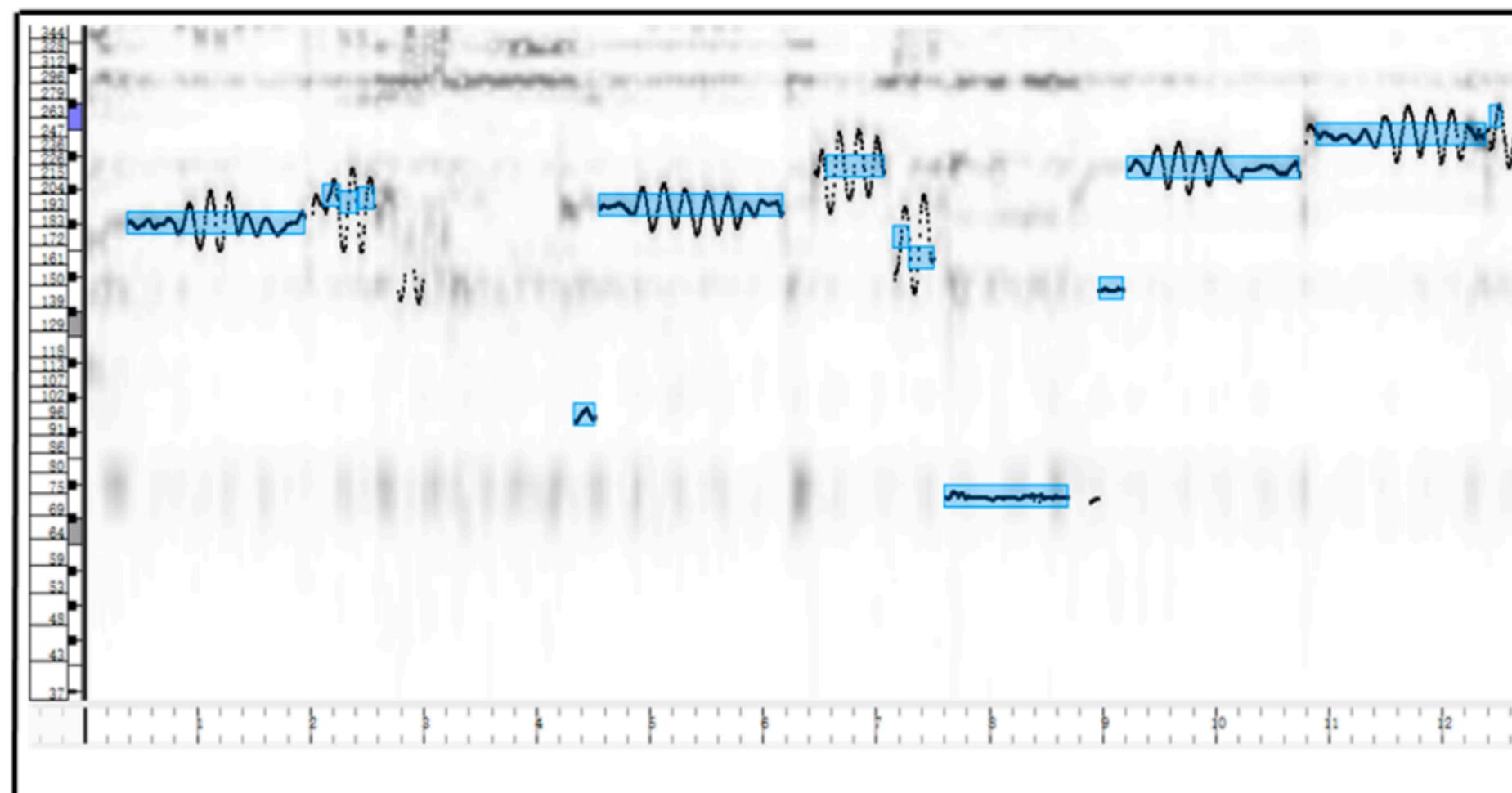
Music Transcription



(a) Frame-level transcription



(c) Stream-level transcription

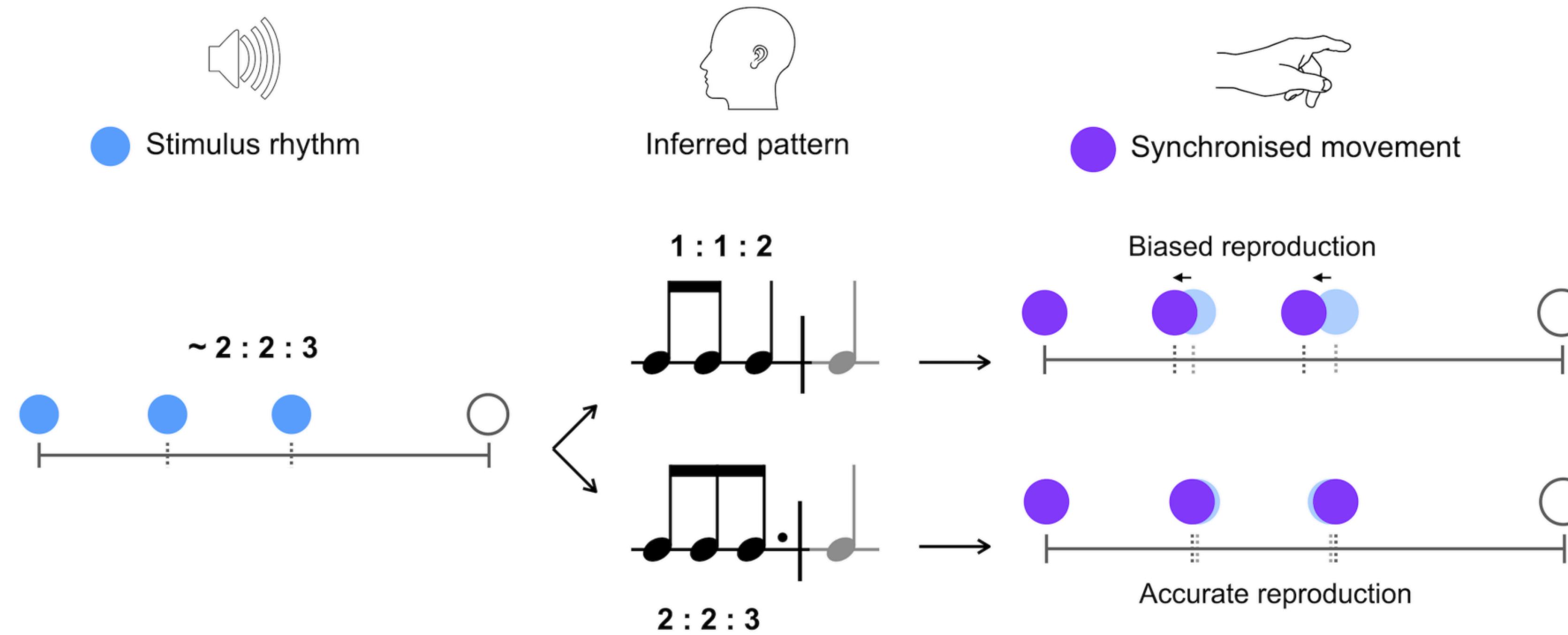


(b) Note-level transcription



(d) Notation-level transcription

Music Cognition: Rhythm Perception



Part 2. Music Information Retrieval Examples

Piano Transcription ~ “ASR”

Music Source Separation ~ “Speech Source Separation”

Beat tracking

Chord recognition

Rhythm: Beat Tracking

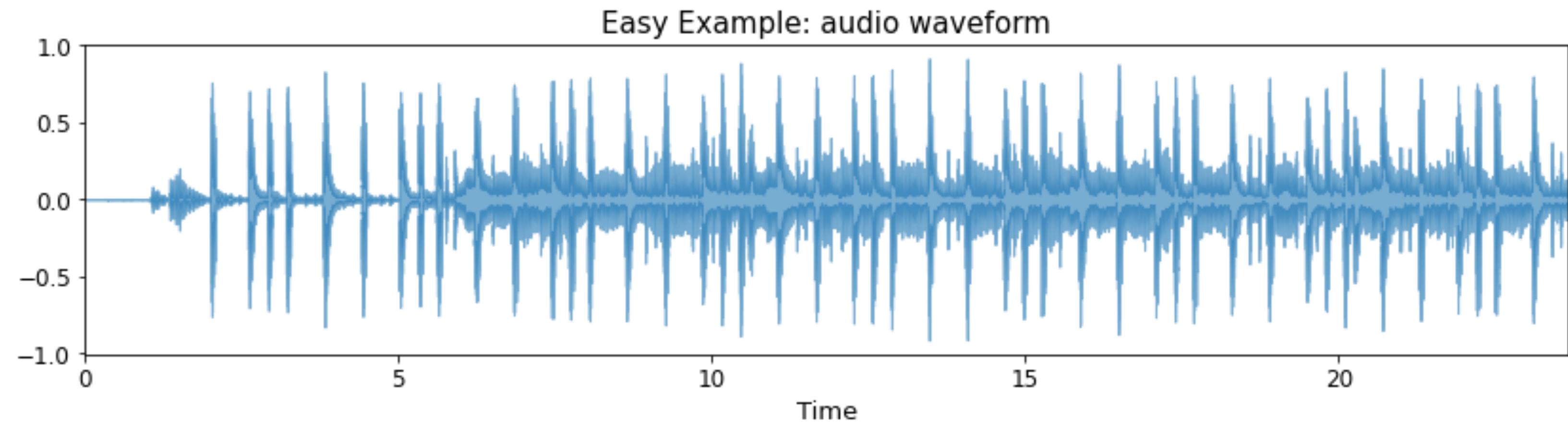
A musical score in bass clef, 3/4 time, with two measures of music. The first measure contains a whole note followed by a dotted half note. The second measure contains a half note followed by a sixteenth-note cluster with grace notes, followed by a eighth-note cluster. Below the score is a horizontal timeline with three rows of dots representing 'tatum', 'beat', and 'downbeat' events.

tatum ● ● ● ● ● ● ● ● ● ● ● ● ● ●

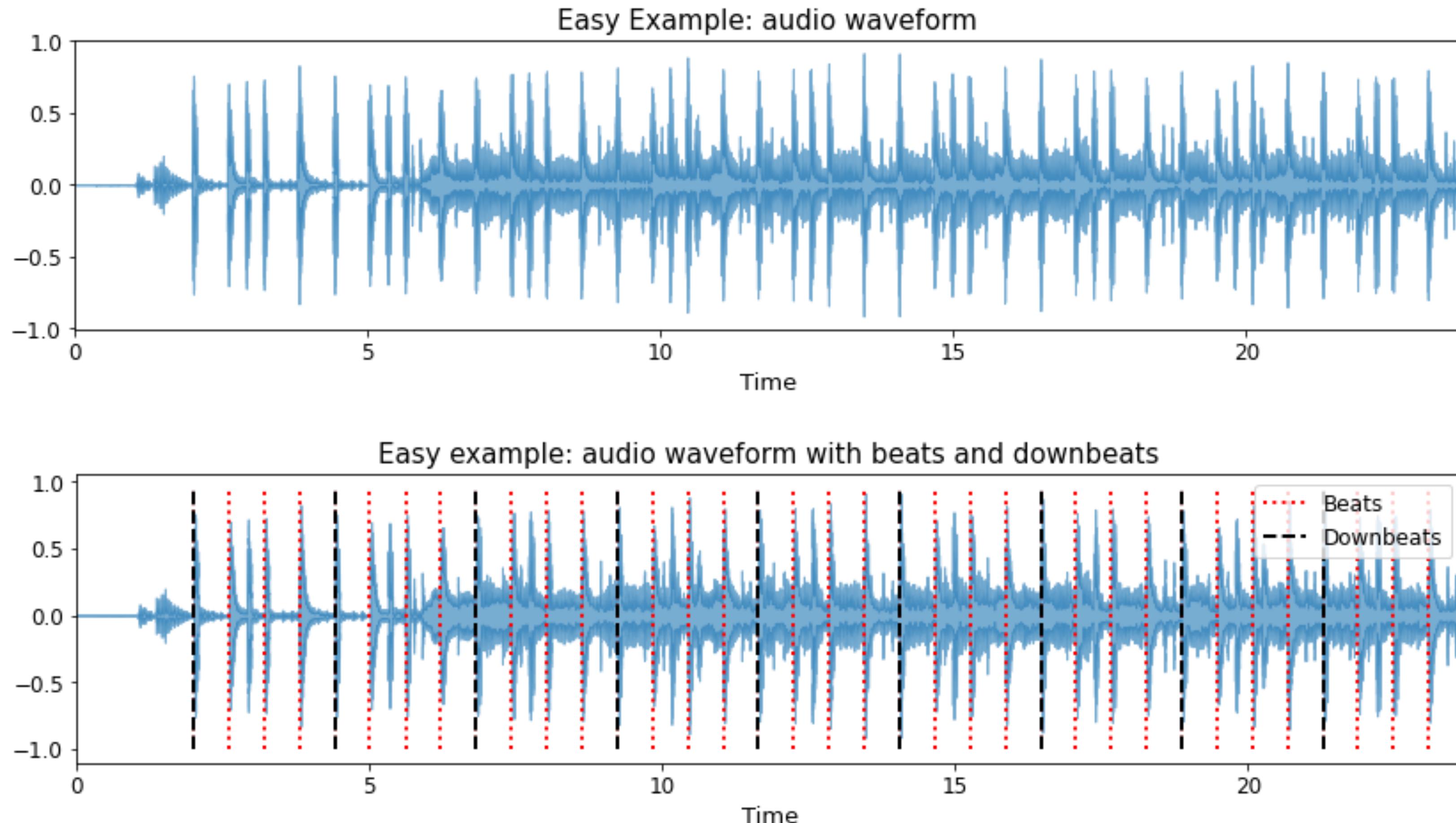
beat ● ● ● ●

downbeat ● ● ● ●

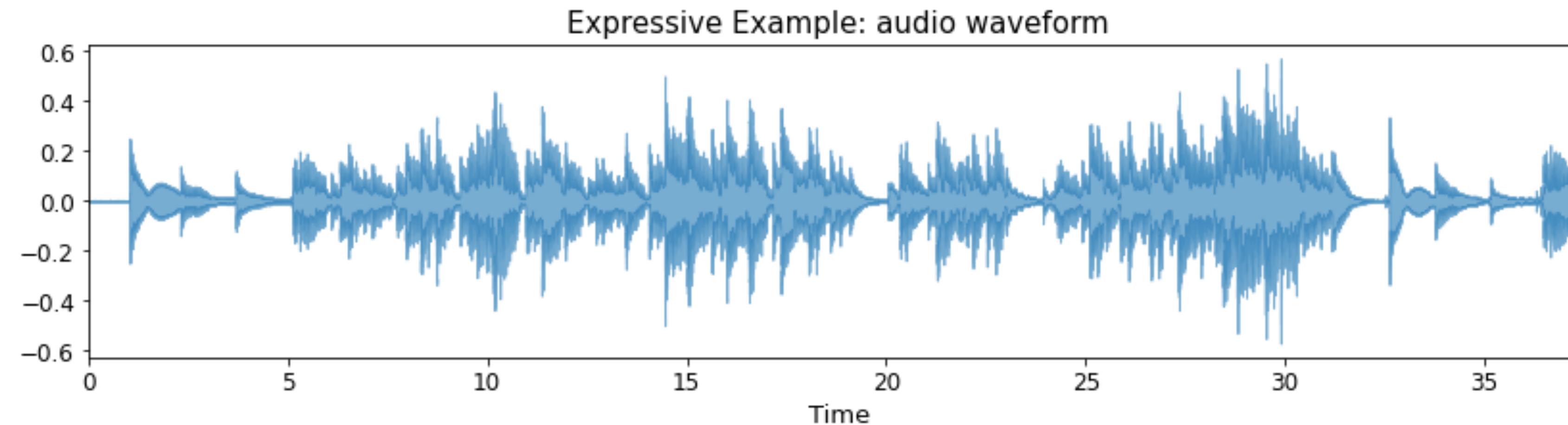
Rhythm: Beat Tracking



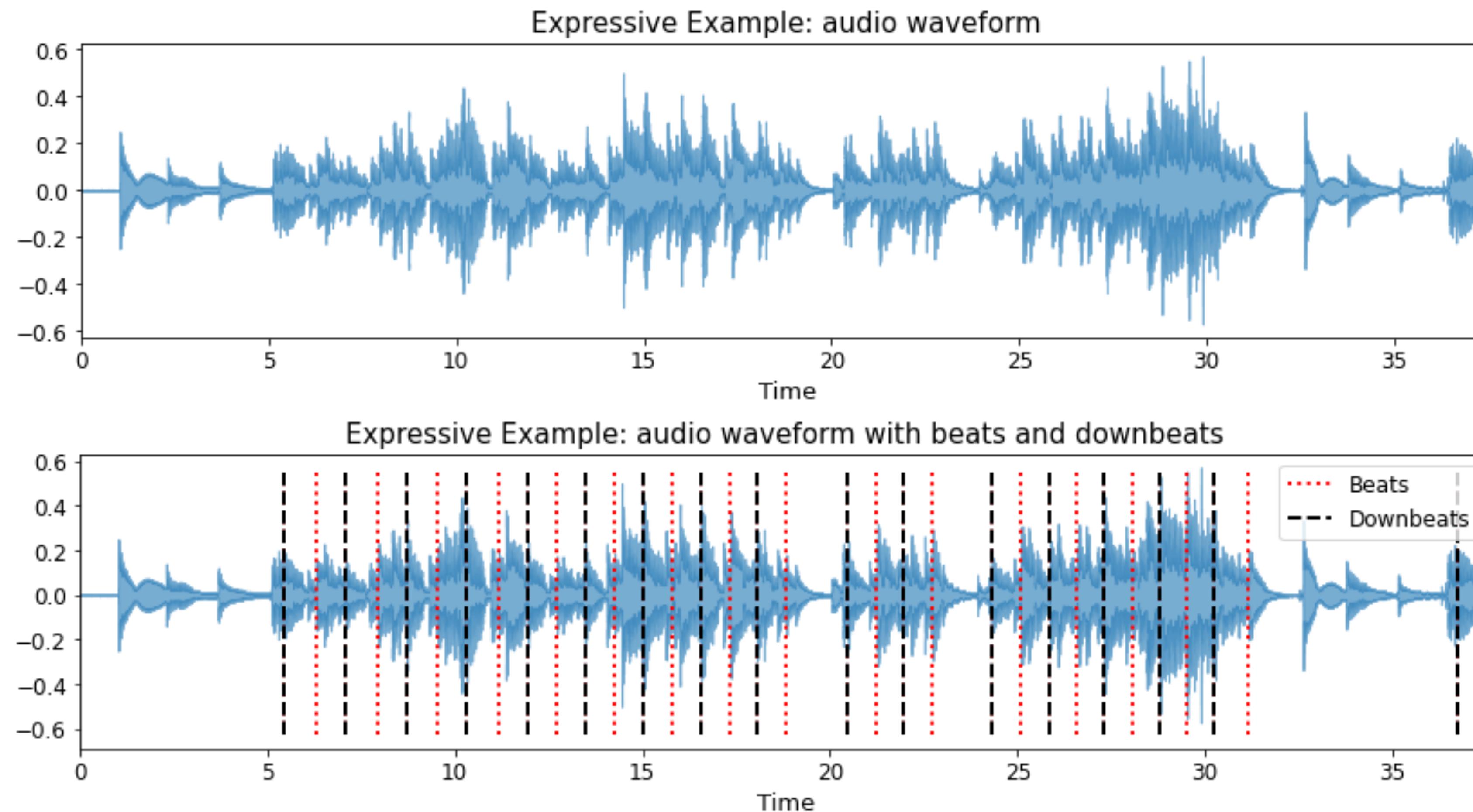
Rhythm: Beat Tracking



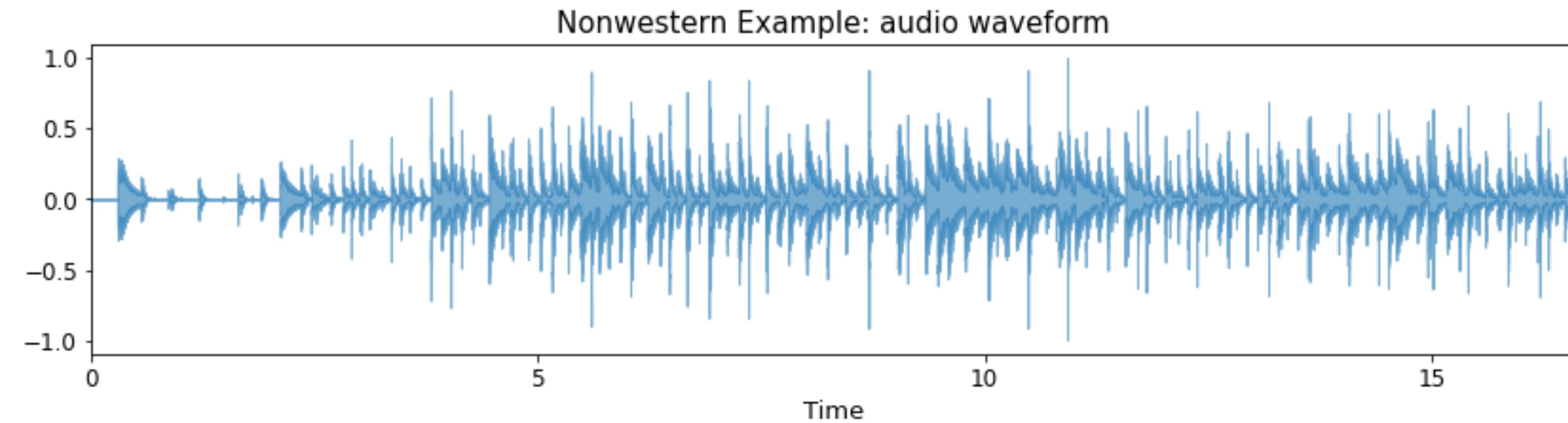
Rhythm: Beat Tracking



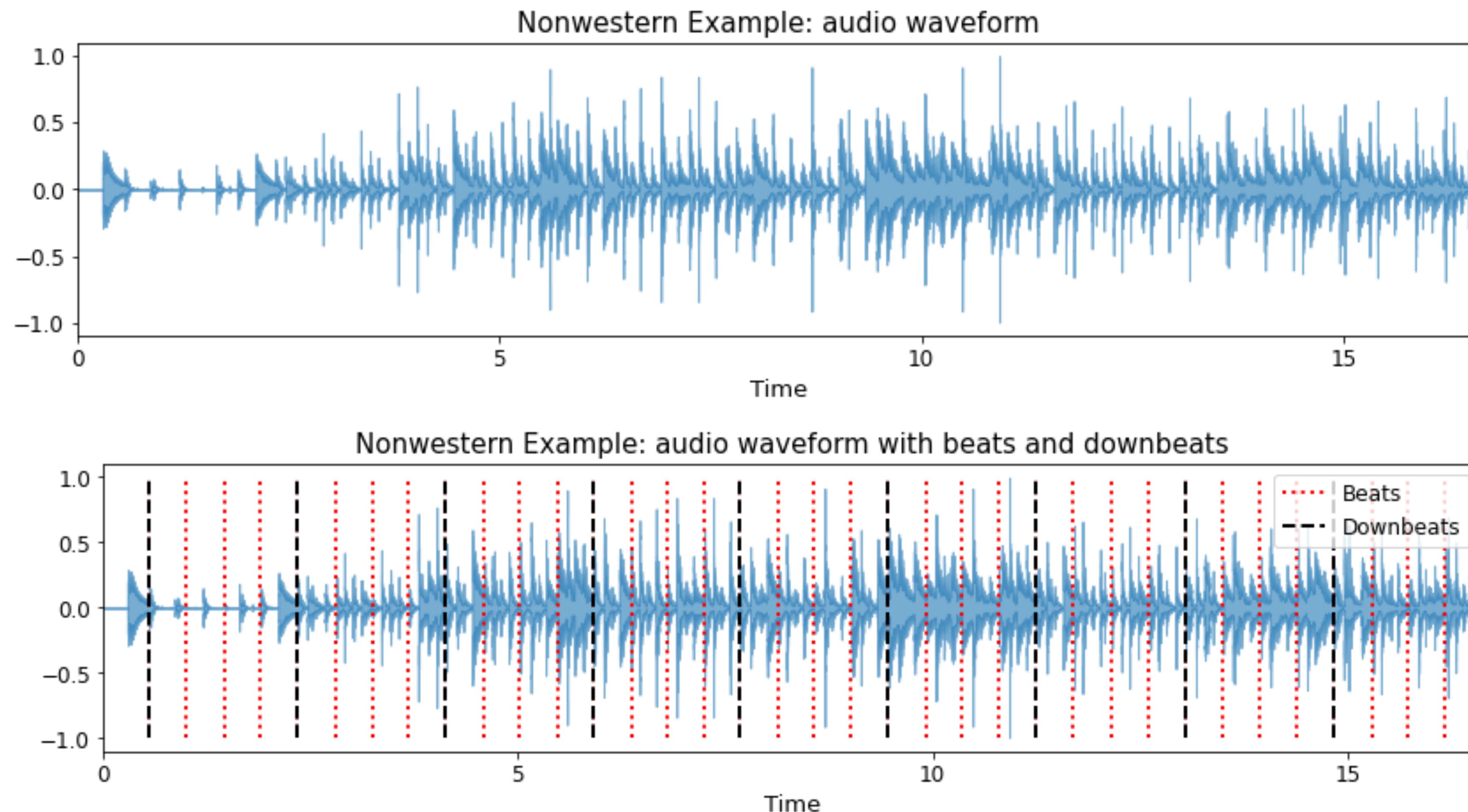
Rhythm: Beat Tracking



Rhythm: Beat Tracking

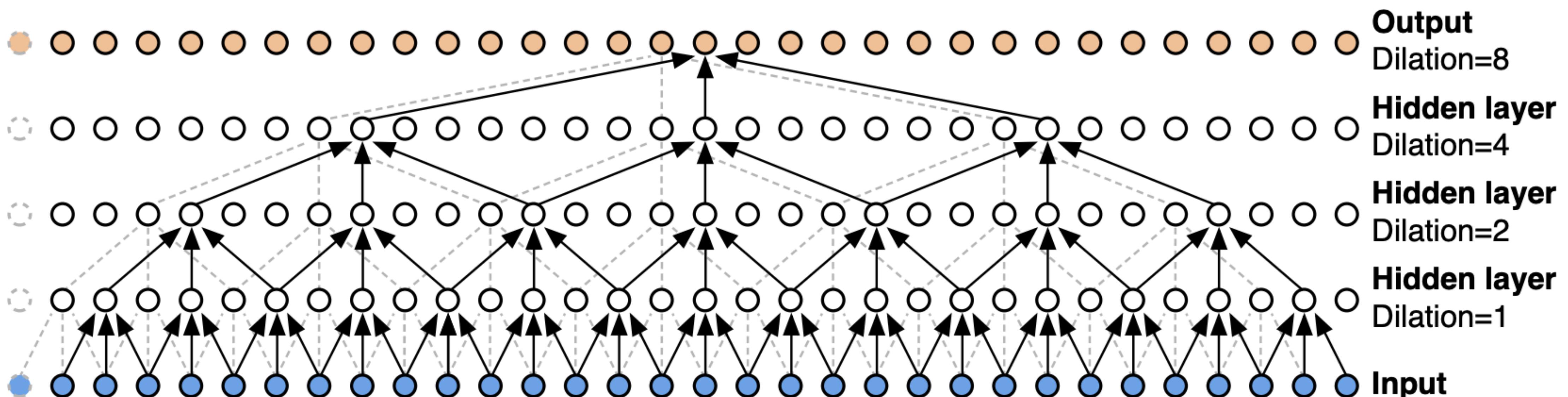


Rhythm: Beat Tracking



Rhythm: Beat Tracking

TCN: Temporal Convolutional Networks + post-processing



Bar Pointer Model

Postprocessing

Neural network output:



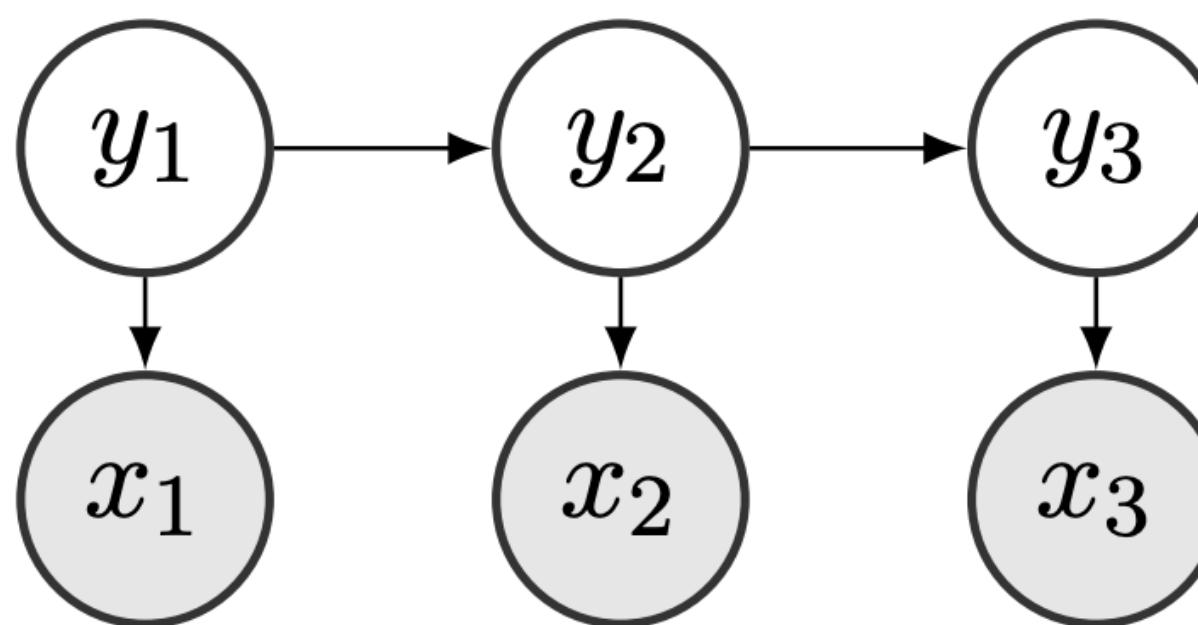
How to convert this to consistent predictions?

Bar Pointer Model

Postprocessing

Dynamic Bayesian Networks:

$$P(\mathbf{y}, \mathbf{x}) = P(\mathbf{y}_1) \prod_{t=2}^T P(\mathbf{y}_t | \mathbf{y}_{t-1}) P(\mathbf{x}_t | \mathbf{y}_t)$$



Bar Pointer Model

Postprocessing

Hidden states: position of a pointer in a bar,

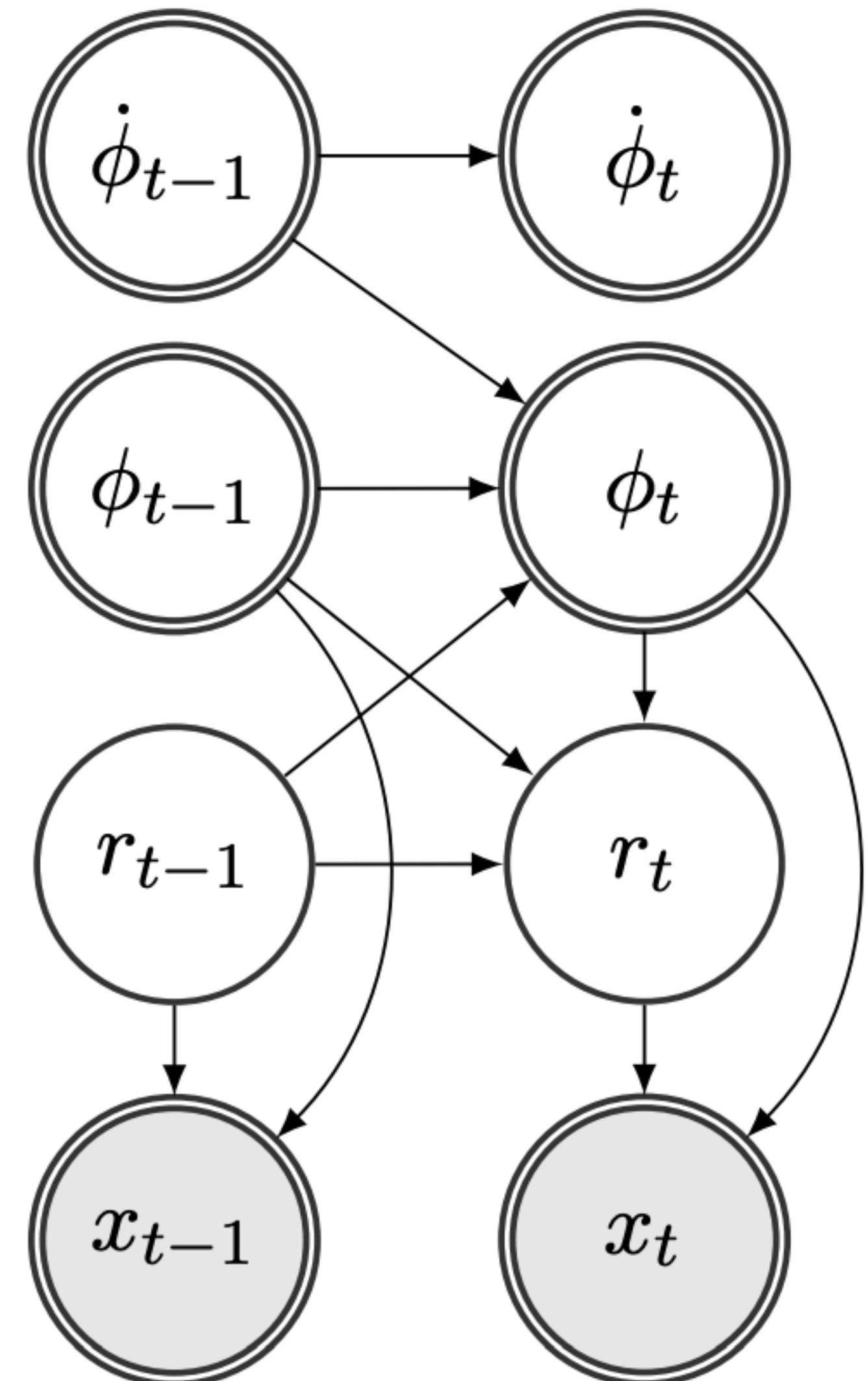
defined by the position, instantaneous tempo and
the rhythmic pattern

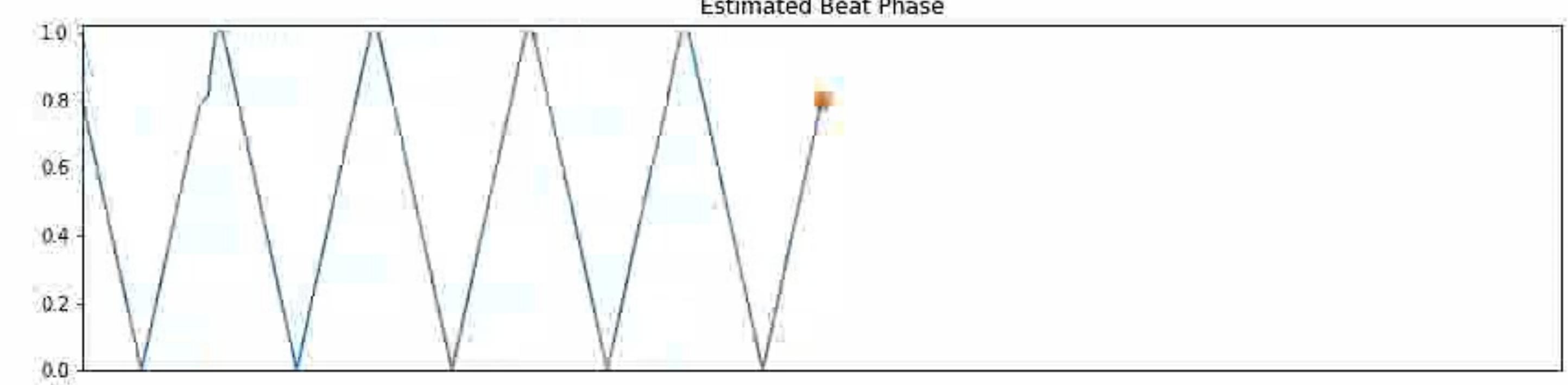
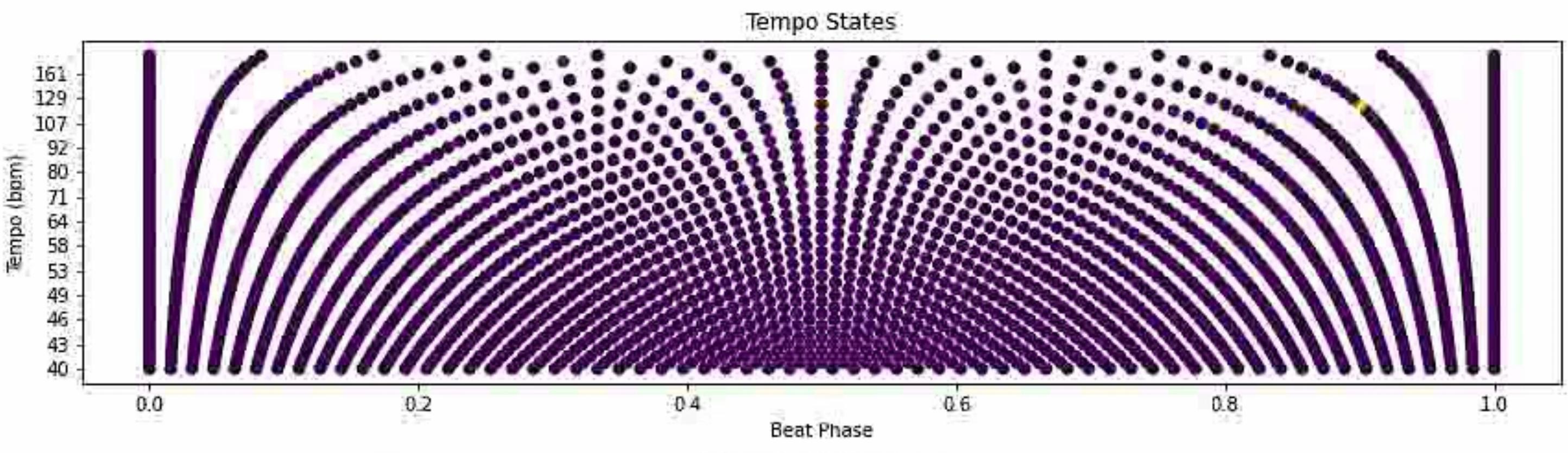
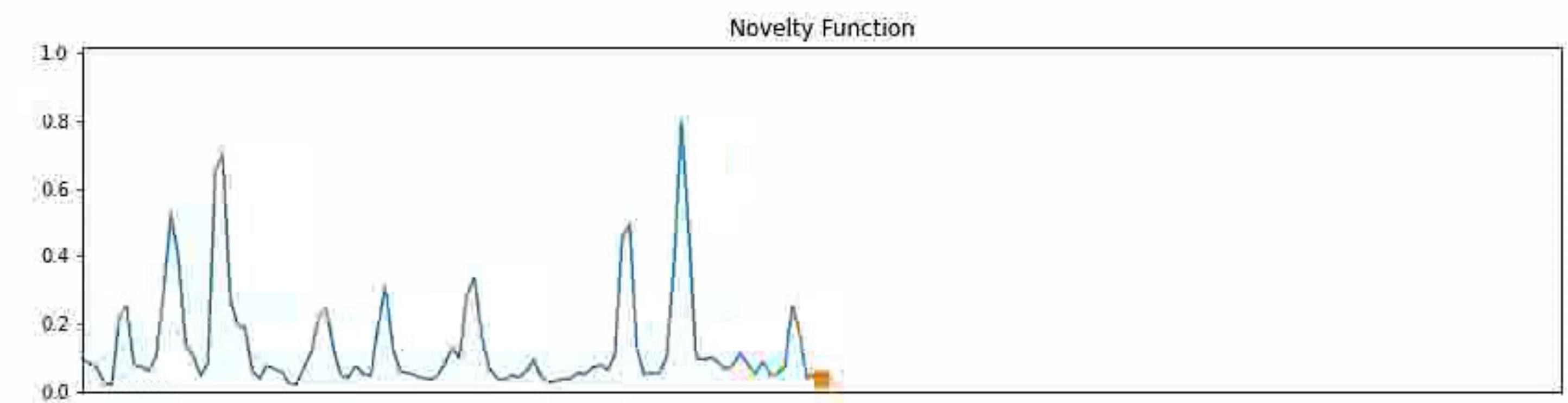
$$P(\mathbf{y}_t | \mathbf{y}_{t-1}) = P(\phi_t | \phi_{t-1}, \dot{\phi}_{t-1}, r_{t-1}) \times P(\dot{\phi}_t | \dot{\phi}_{t-1}) \times P(r_t | r_{t-1}, \phi_{t-1}, \phi_t)$$

$$P(\phi_k | \phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) = \mathbb{1}_\phi$$

$$P(\dot{\phi}_k | \dot{\phi}_{k-1}) \propto \mathcal{N}(\dot{\phi}_{k-1}, \sigma_{\dot{\phi}}^2) \times \mathbb{1}_{\dot{\phi}}$$

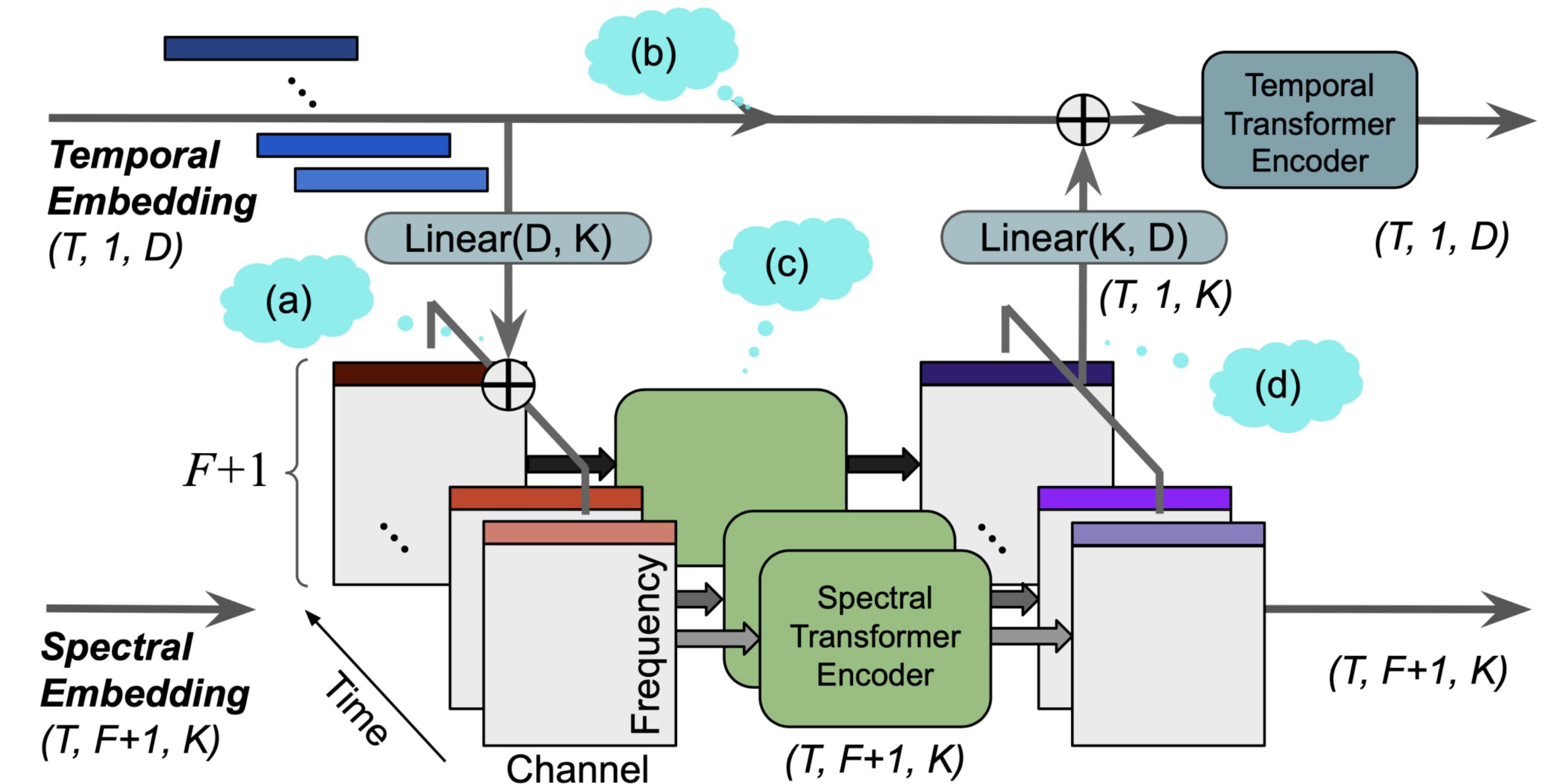
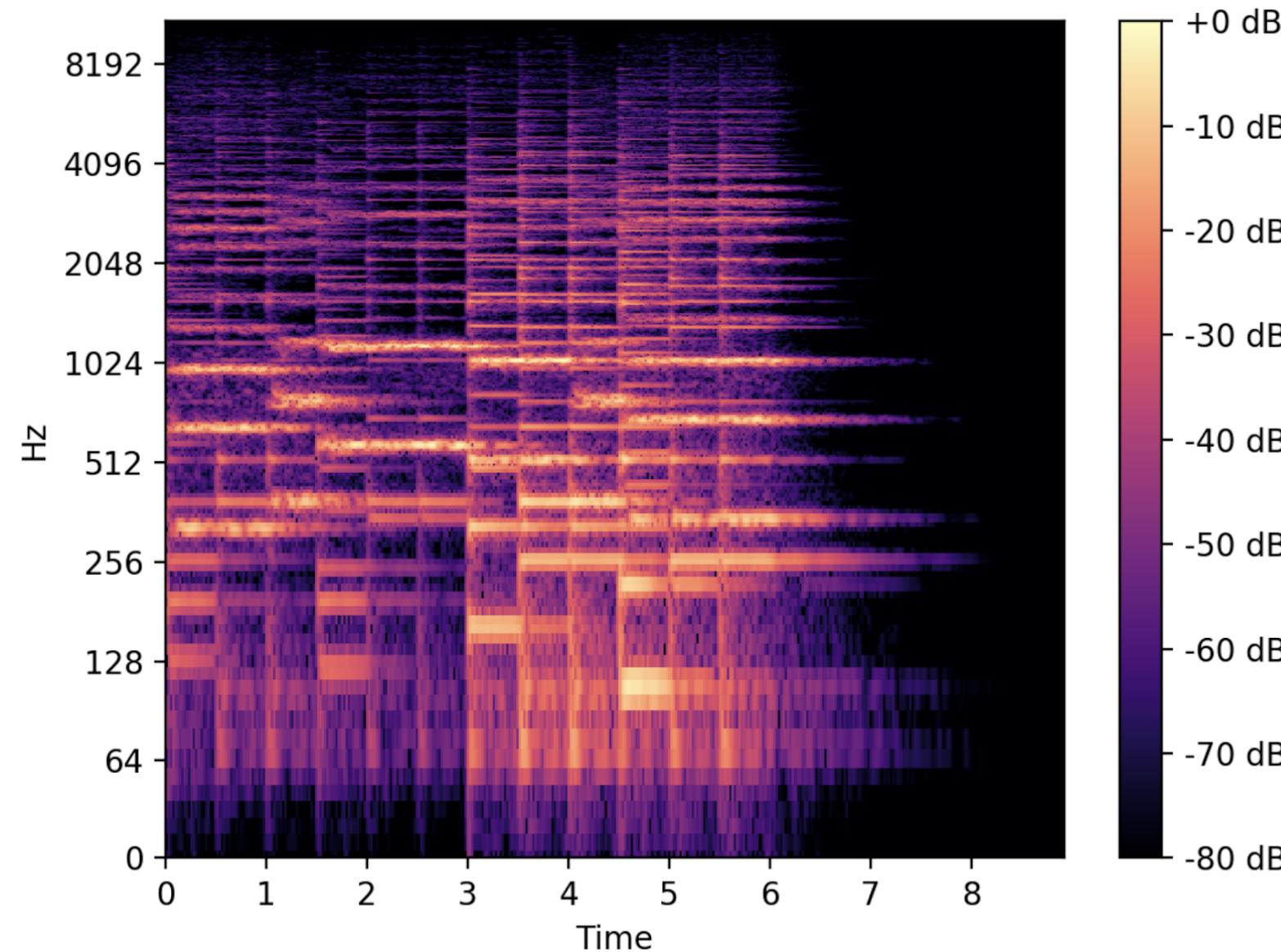
$$P(r_k | r_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbf{A}(r_{k-1}, r_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_r & \text{else} \end{cases}$$





SpecTNT

Transformers for spectrograms



Exploits both **time** and **frequency** dependencies

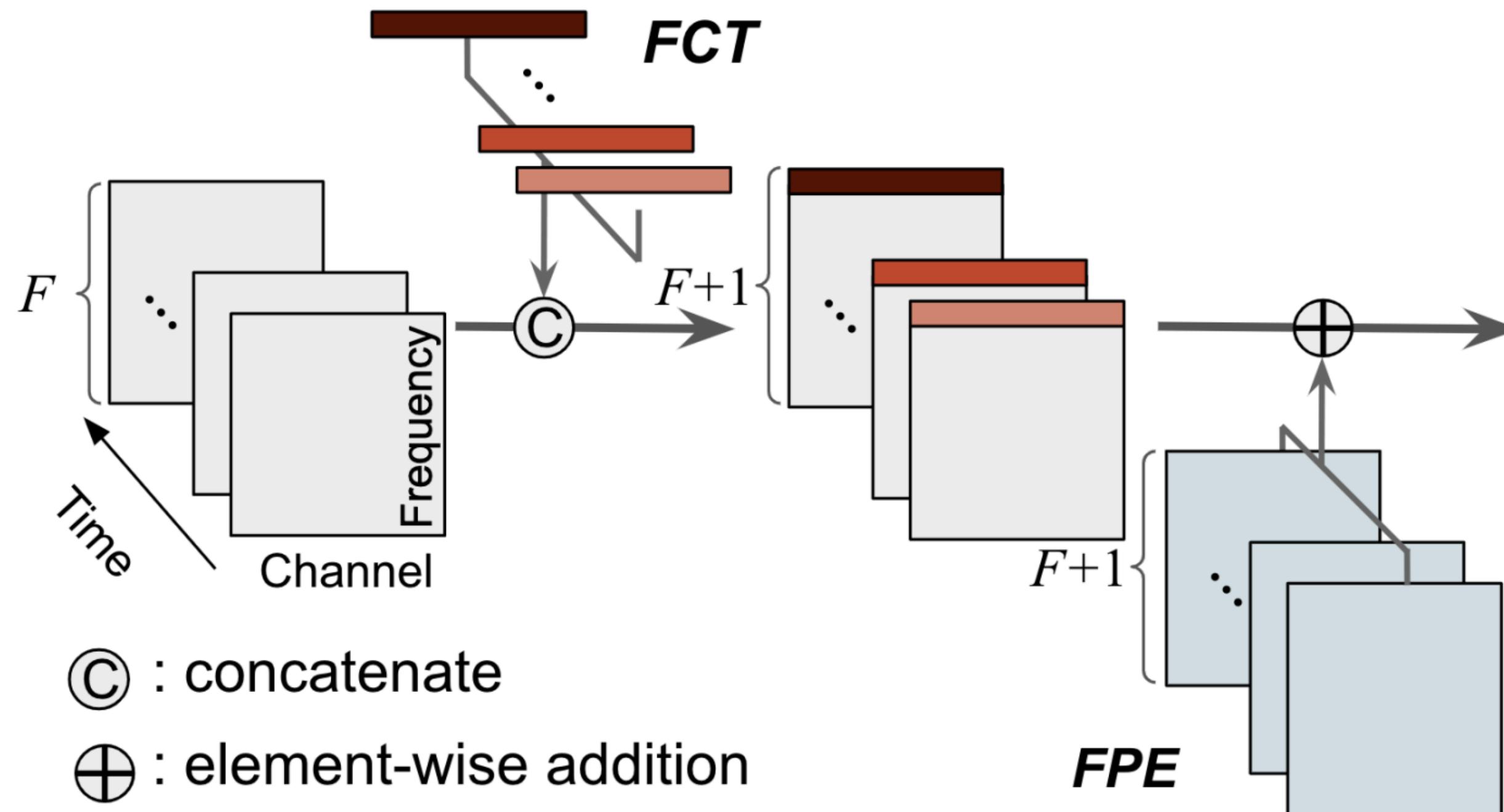
Time Transformer: Captures temporal dependencies along the time axis

Frequency Transformer: Captures harmonic and spectral relationships along the frequency axis.

Alternating interaction between Time and Frequency blocks.

SpecTNT

Transformers for spectrograms



Exploits both **time** and **frequency** dependencies

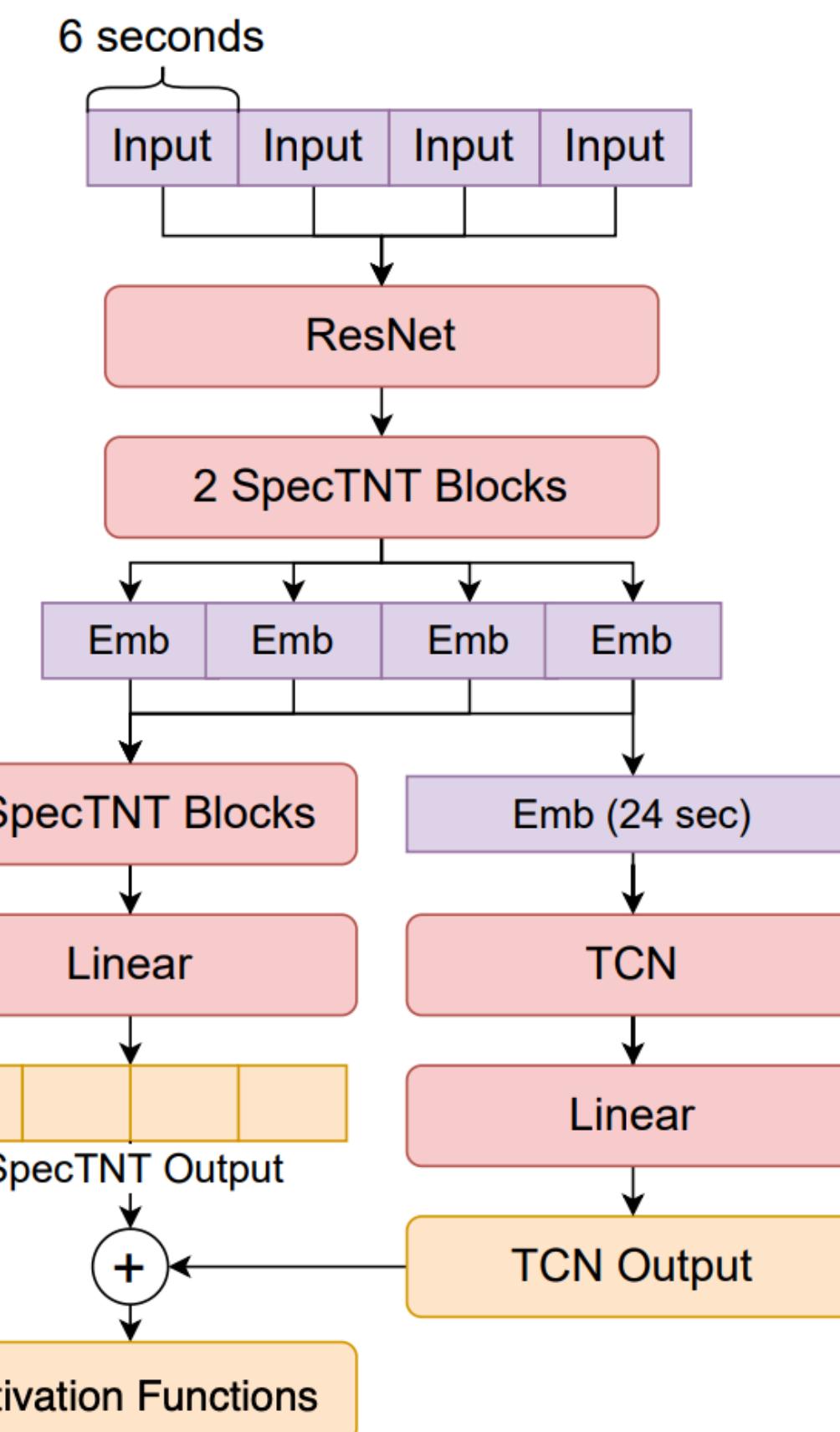
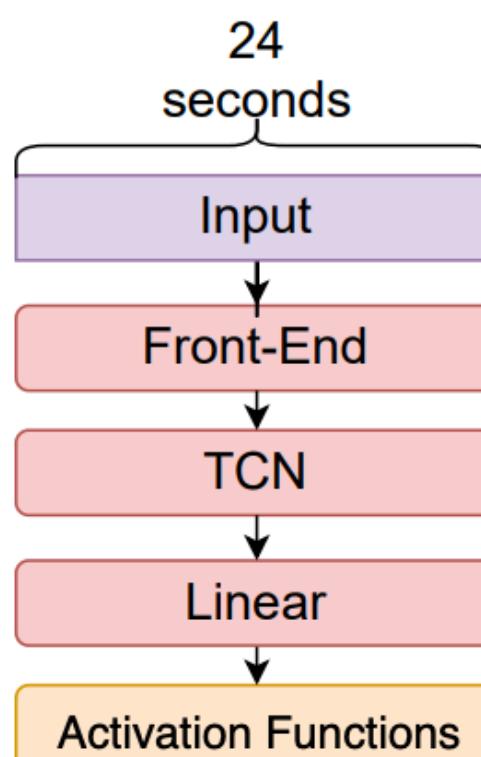
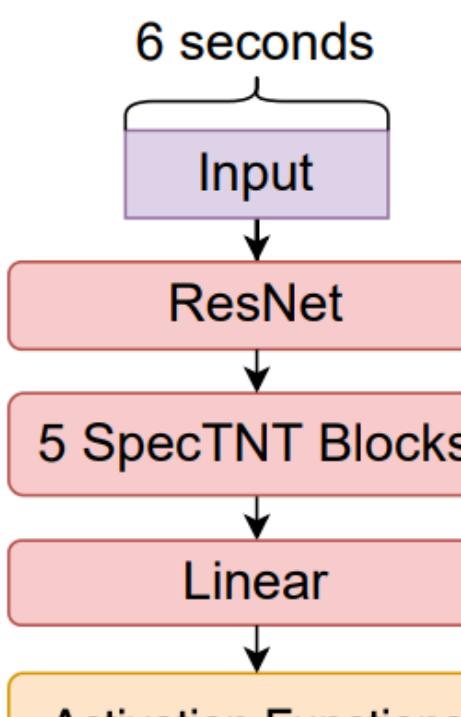
Time Transformer: Captures temporal dependencies along the time axis

Frequency Transformer: Captures harmonic and spectral relationships along the frequency axis.

Alternating interaction between Time and Frequency blocks.

Rhythm: Beat Tracking

SpecTNT-TCN



	F1	CMLt	AMLt	F1	CMLt	AMLt
<i>RWC-POP</i>				<i>Harmonix Set</i>		
Böck et al. [18]	.943	-	-	.933 [†]	.841 [†]	.938 [†]
TCN (baseline)	.947	.922	.952	.946	.895	.942
SpecTNT	.953	.925	.957	.947	.896	.943
SpecTNT-TCN	.950	.925	.958	.953	.939	.959
<i>SMC</i>				<i>Beatles</i>		
Böck et al. [18]	.516	.406	.575	.918	-	-
Böck et al. [17]	.544	.443	.635	-	-	-
TCN (baseline)	.560	.474	.621	.933	.870	.933
SpecTNT	.602*	.515*	.661	.940	.898	.929
SpecTNT-TCN	.605*	.514*	.663	.943	.896	.938
<i>Ballroom</i>				<i>Hainsworth</i>		
Davies et al. [15]	.933	.881	.929	.874	.795	.930
Böck et al. [17]	.962	.947	.961	.902	.848	.930
TCN (baseline)	.940	.870	.957	.860	.849	.915
SpecTNT	.927	.856	.939	.866	.865	.914
SpecTNT-TCN	.962*	.939*	.967	.877	.862	.915

Harmony: Chord Recognition

Interval definition

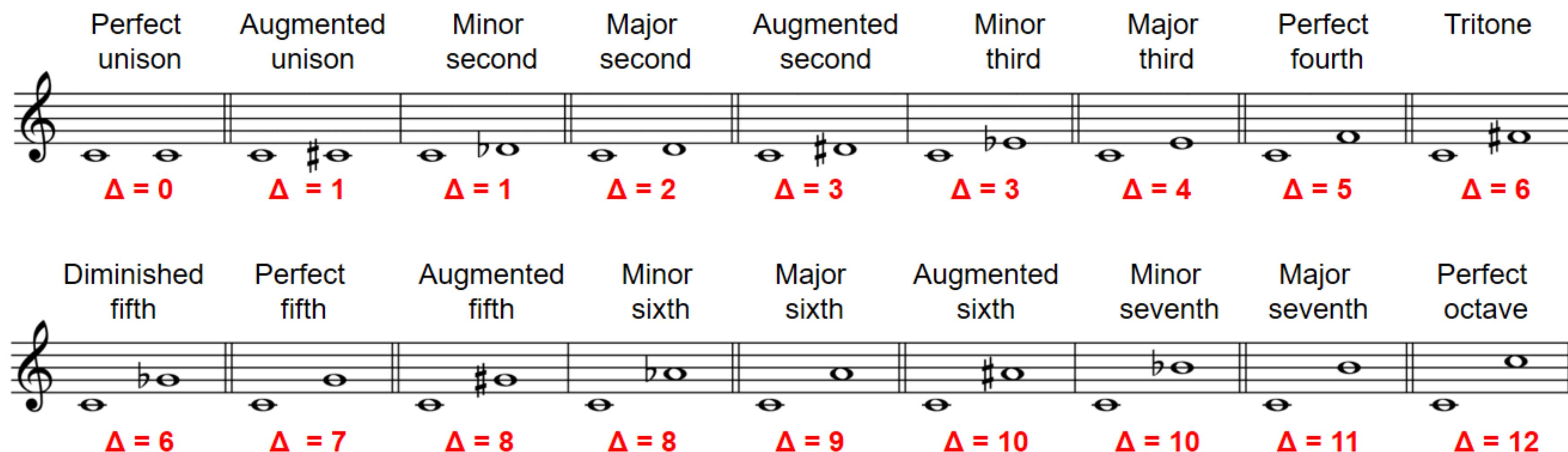


Figure 5.2b from [Müller, FMP, Springer 2015]

Harmony: Chord Recognition

Intervals in harmonic series

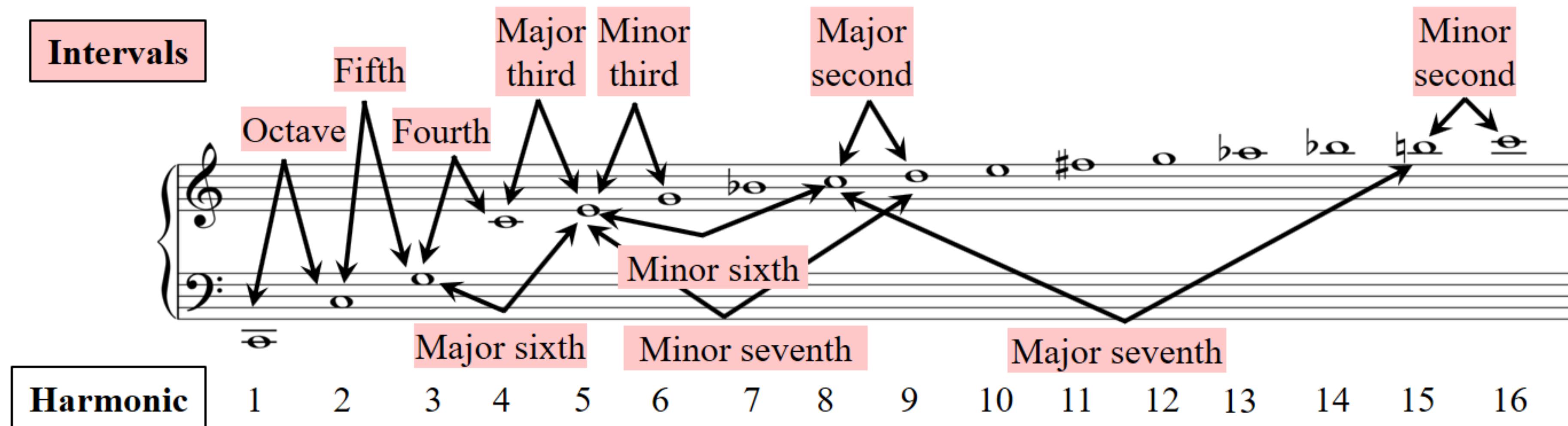
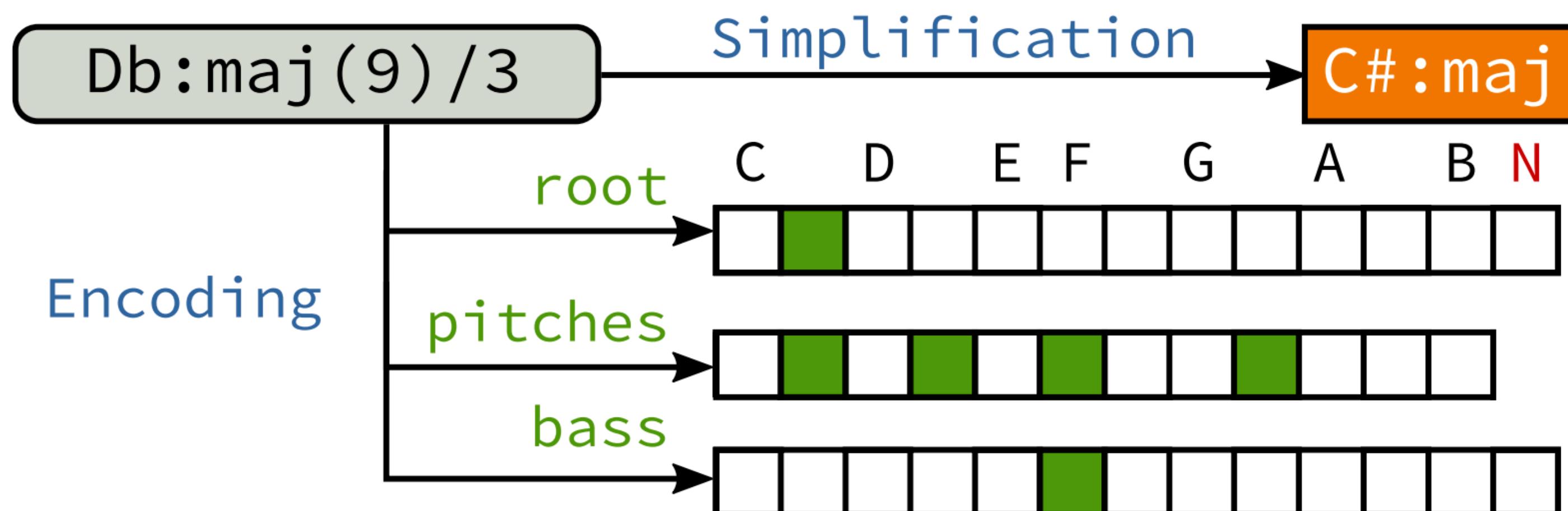


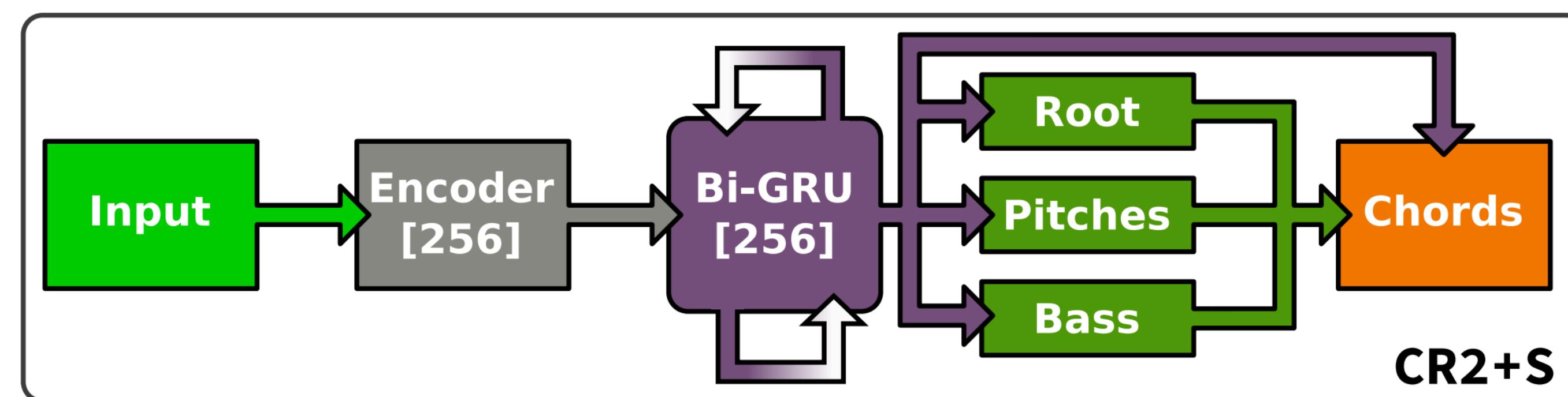
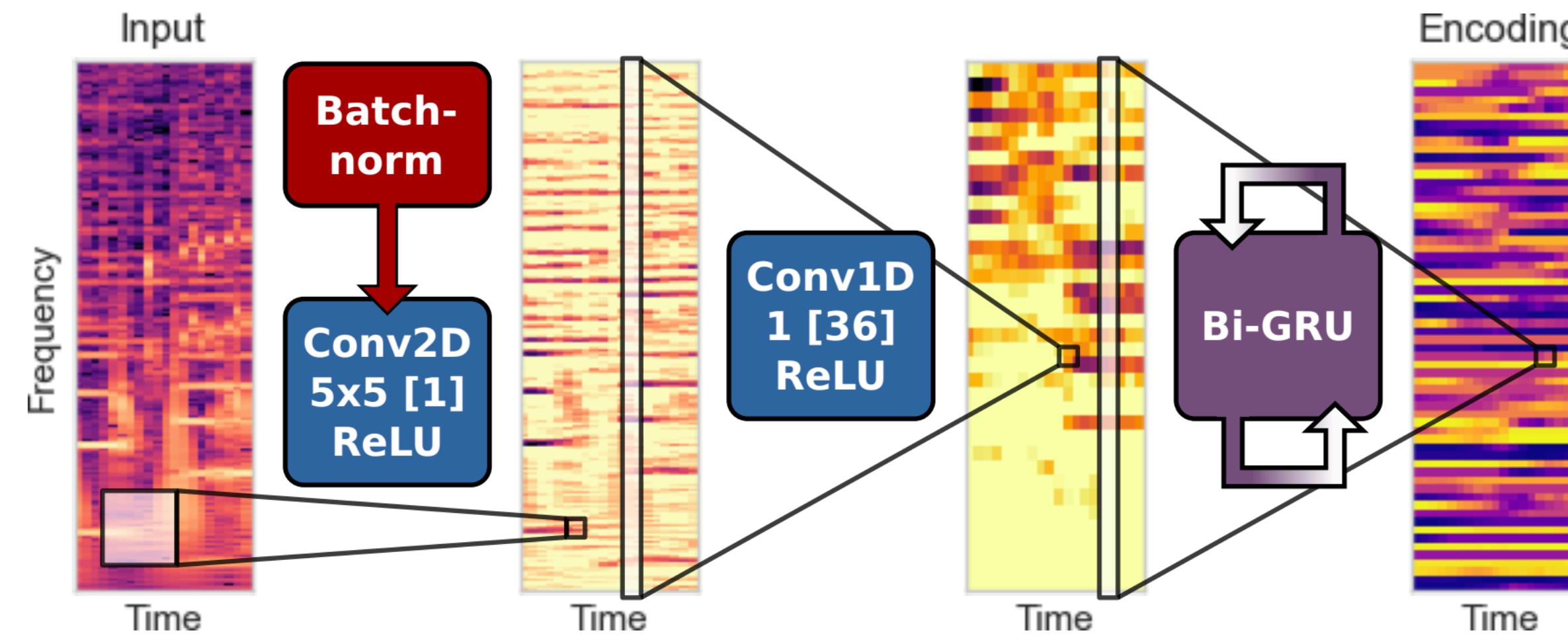
Figure 5.4 from [Müller, FMP, Springer 2015]

Harmony: Chord Recognition

Key features: large vocabulary, “heavy-tail” distribution, annotations are sparse and ambiguous



Harmony: Chord Recognition

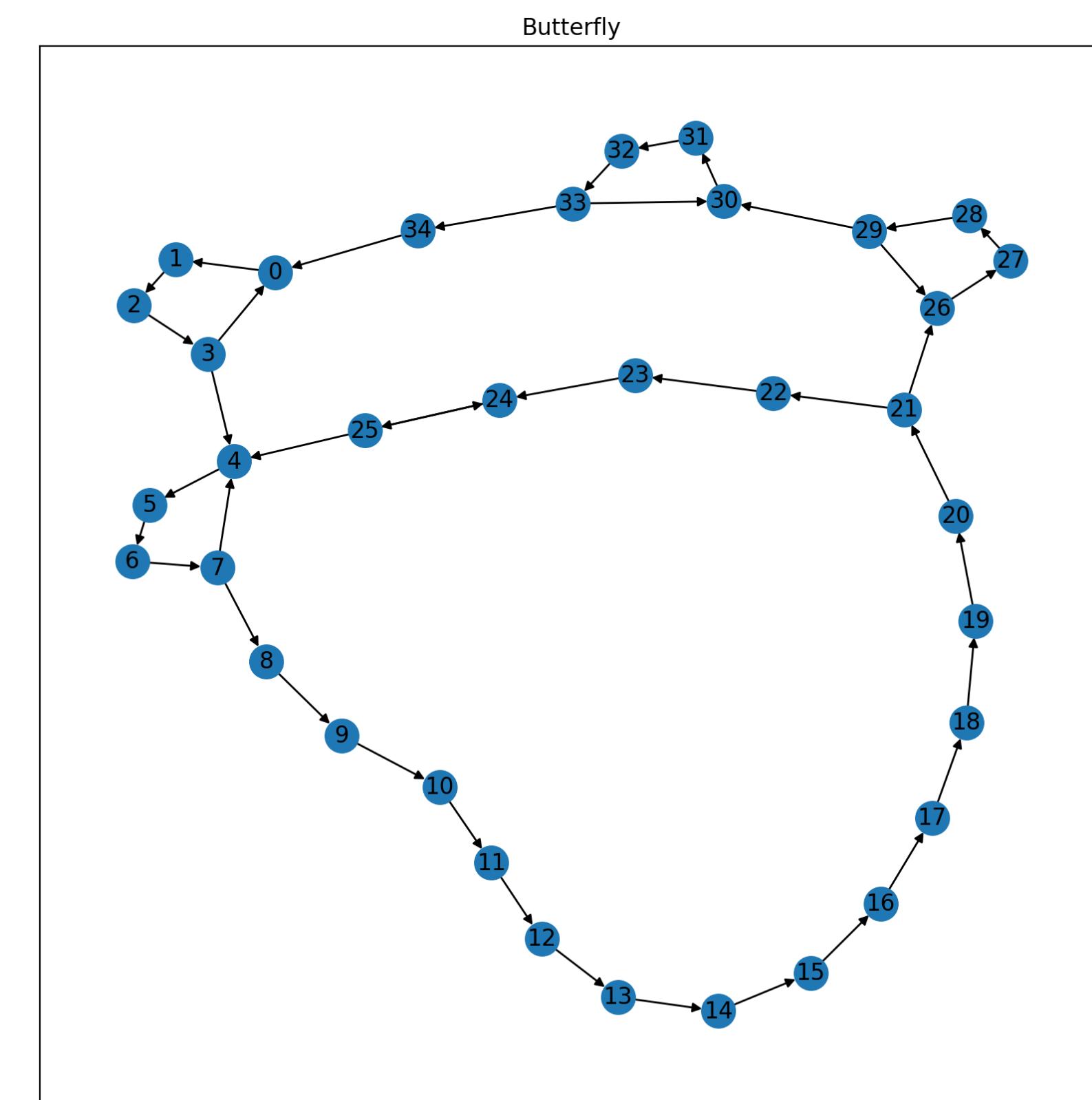


Lead Sheet

- Contains necessary information regarding musical structure
 - Represents a graphical structure
 - Sometimes also contains a melody
 - Large collections of lead sheets:
 - iRealPro Jazz 1410
 - Wikifonia
 - Band in a Box (BIAB)

Butterfly

The diagram illustrates a directed graph structure, specifically a butterfly graph, which is a common representation in music theory. It consists of 34 numbered nodes (0 through 33) arranged in two main horizontal layers. The top layer contains nodes 0, 3, 4, 23, 24, 25, 22, 30, 31, 32, 33, and 34. The bottom layer contains nodes 1, 2, 5, 6, 7, 8, 9, 10, 11, and 12. Directed edges connect nodes between adjacent layers: (0,1), (0,3), (1,2), (3,4), (4,5), (4,7), (5,6), (6,7), (7,8), (8,9), (9,10), (10,11), (11,12), (30,31), (31,32), (32,33), (33,34), (23,24), (24,25), (25,22), and (22,30). There are also internal connections within the layers: (3,0), (3,3), (4,0), (4,3), (4,4), (4,23), (7,4), (7,8), (8,4), (8,24), (9,8), (9,25), (10,9), (10,24), (11,10), and (11,23).



A lead sheet and its graph

Lead Sheet Alignment

- Task definition:
 - Match each moment of musical audio with a position in a lead sheet.
- If solved, it would provide:
 - A local harmonic context to the result of AMT
 - A “global” structural context needed to analyse solos
 - A form-aware theme/solo segmentation solution

Lead Sheet Alignment SotA

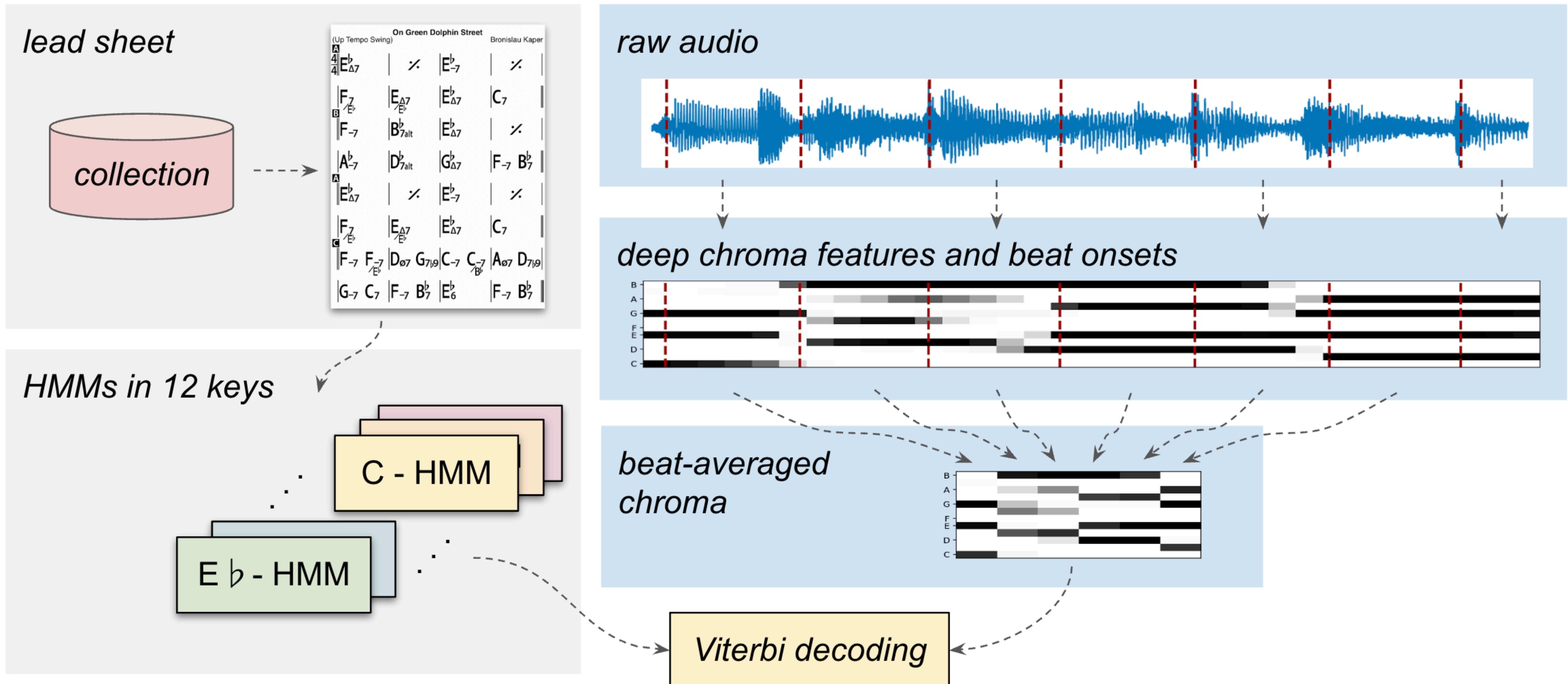
- Current solution (ours, WASPAA 2023):
 - Extract audio features with CRNN
 - Compile an HMM using a lead sheet and use Viterbi decoding
 - Observations: CRNN deep chroma features
 - Transition probabilities: a lead sheet graph connectivity
 - Emission probabilities:

$$P(Y_a | X_{ls}) := \frac{\sigma(\text{Hamming}(X_{ls}, \hat{Y}_a))}{\sum_{Y \in \{0,1\}^{12}} \sigma(\text{Hamming}(X_{ls}, Y))},$$

where X_{ls} is a chroma of a chord in a lead sheet,

Y_a - 12-dim CRNN output, \hat{Y}_a - binarised CRNN output (threshold 0.5)

Lead Sheet Alignment



(Up Tempo Swing)

Giant Steps

John Coltrane

$\frac{4}{4}$ | B_{D7} D₇ | G_{D7} B₇^b | E_{D7}^b | A₋₇ D₇ |
G_{D7} B₇^b	E_{D7}^b F₇[#]	B_{D7}	F₋₇ B₇^b	
E_{D7}^b	A₋₇ D₇	G_{D7}	C₋₇[#] F₇[#]	
B_{D7}	F₋₇ B₇^b	E_{D7}^b	C₋₇[#] F₇[#]	

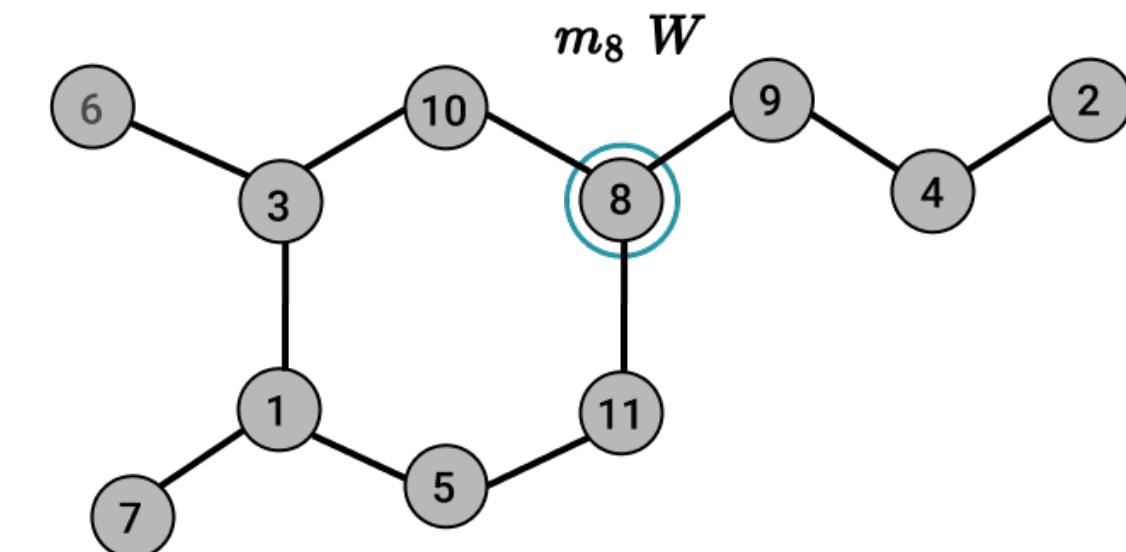
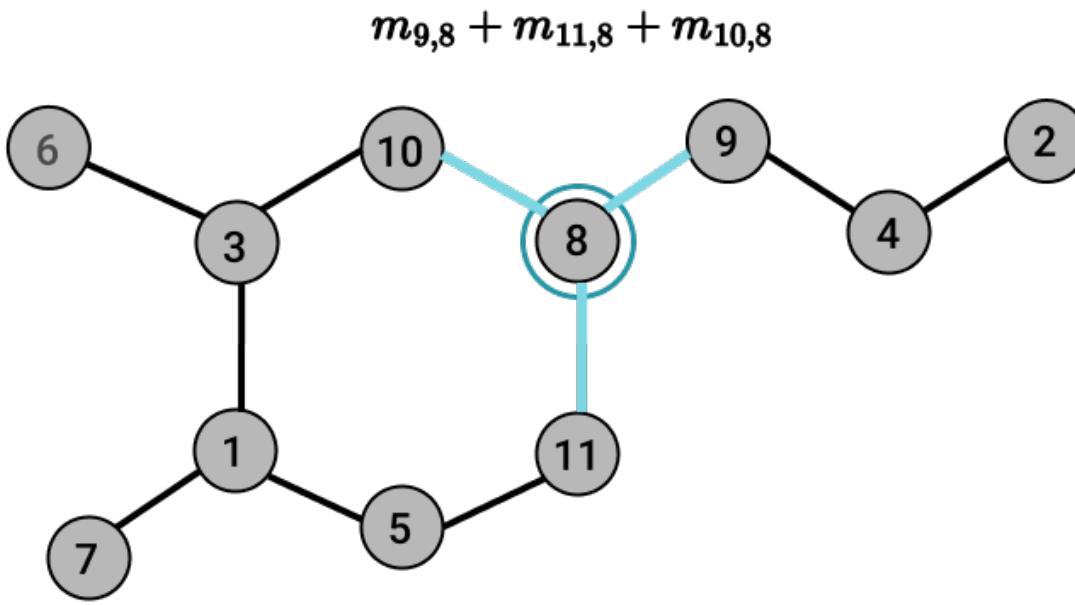
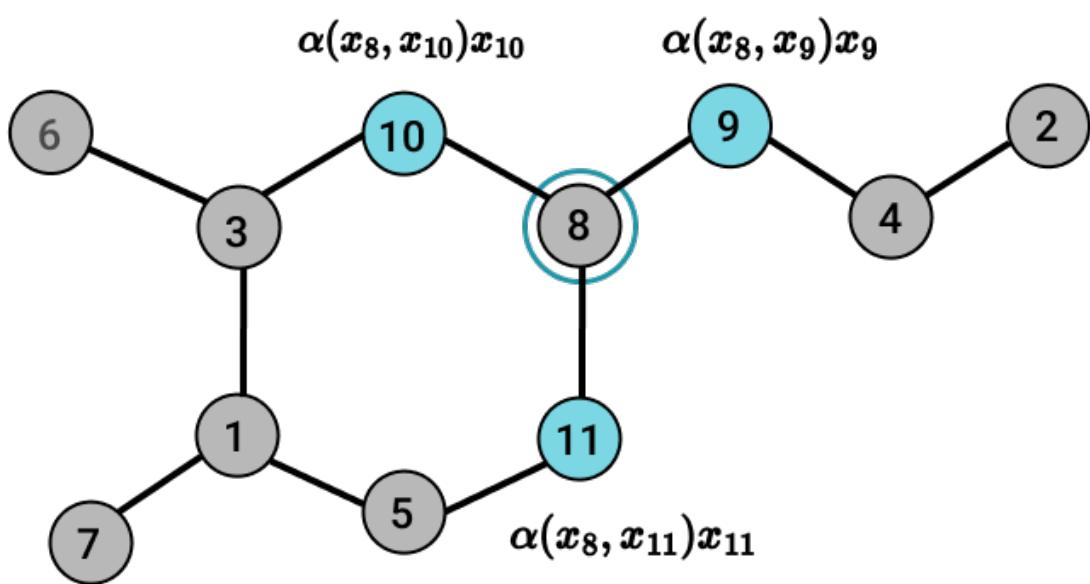


Bi-directional Neighbour Attention

$$f_{msg}(x_i, x_j) = \alpha(x_i, x_j)x_j$$

$$f_{agg}(\{m_i\}) = \sum_j (m_{ij})$$

$$f_{upd}(x_i, m_i) = m_i W$$



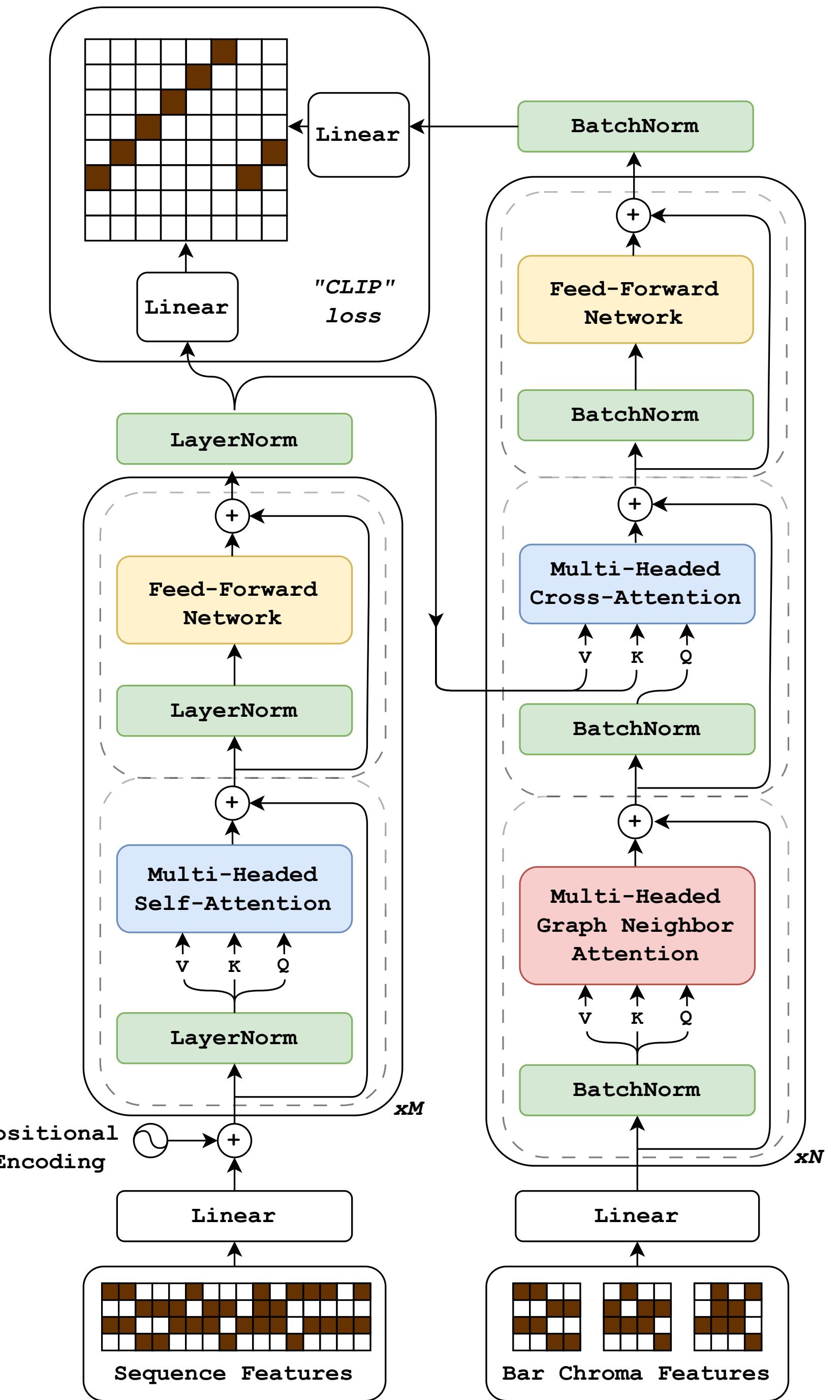
- Neighbor attention:

$$f_{msg}^F(x_i, x_j) = \sigma(q(x_i)^T k(x_j))v(x_j),$$

$$f_{msg}^R(x_i, x_j) = \sigma(\hat{q}(x_j)^T \hat{k}(x_i))\hat{v}(x_i)$$

Lead Sheet Transformer

- Sequence Encoder
 - A standard transformer encoder cell
- Graph Encoder
 - Neighbour attention block (bi-directional)
 - Graph/sequence Cross-attention
 - Feed-forward block
- Contrastive loss
 - Symmetric cross-entropy, “CLIP”



Part 3: Music Generation

JukeBox (OpenAI)

MusicGEN (Meta)

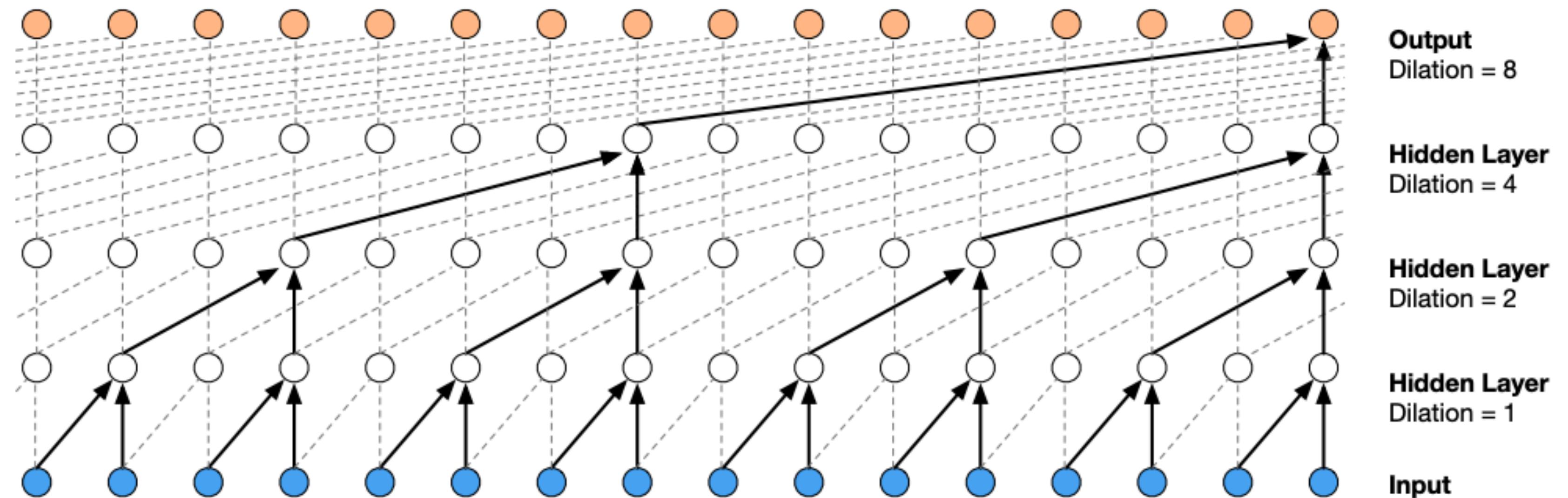
MusicLM (Google)

Waveform Generation

WaveNet / SampleRNN

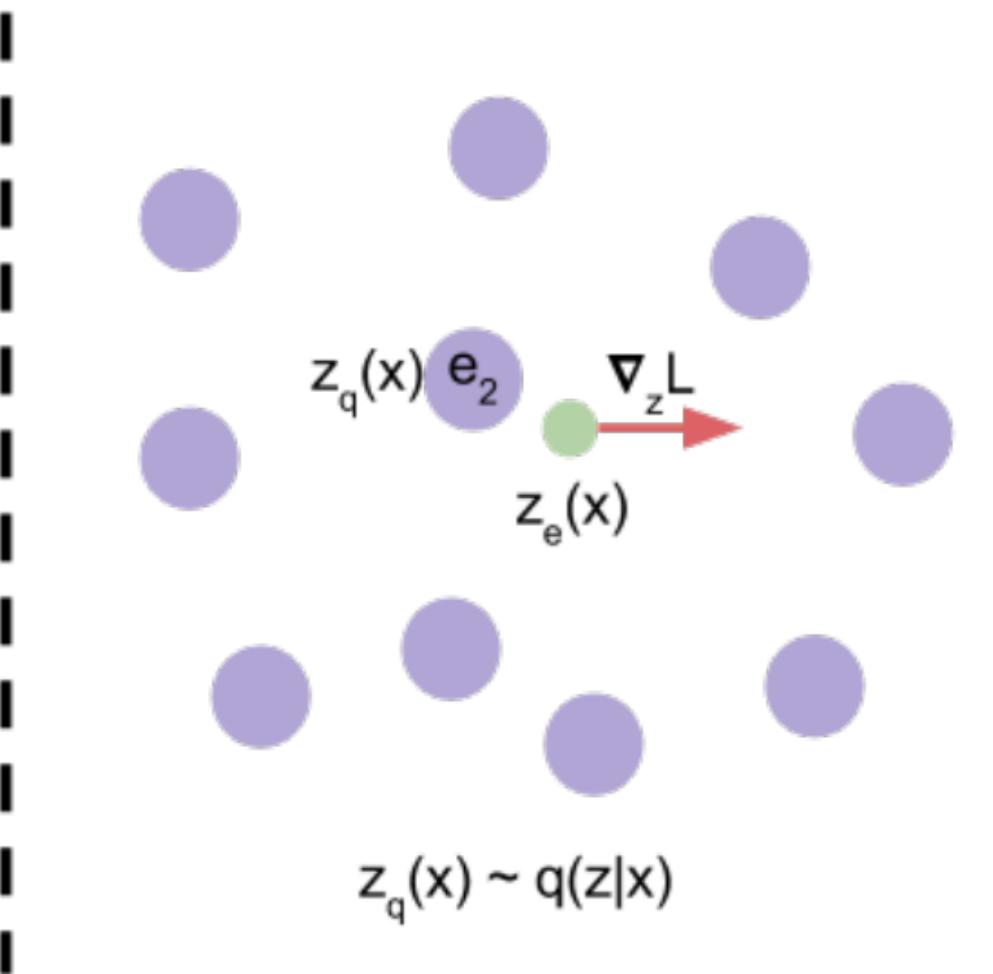
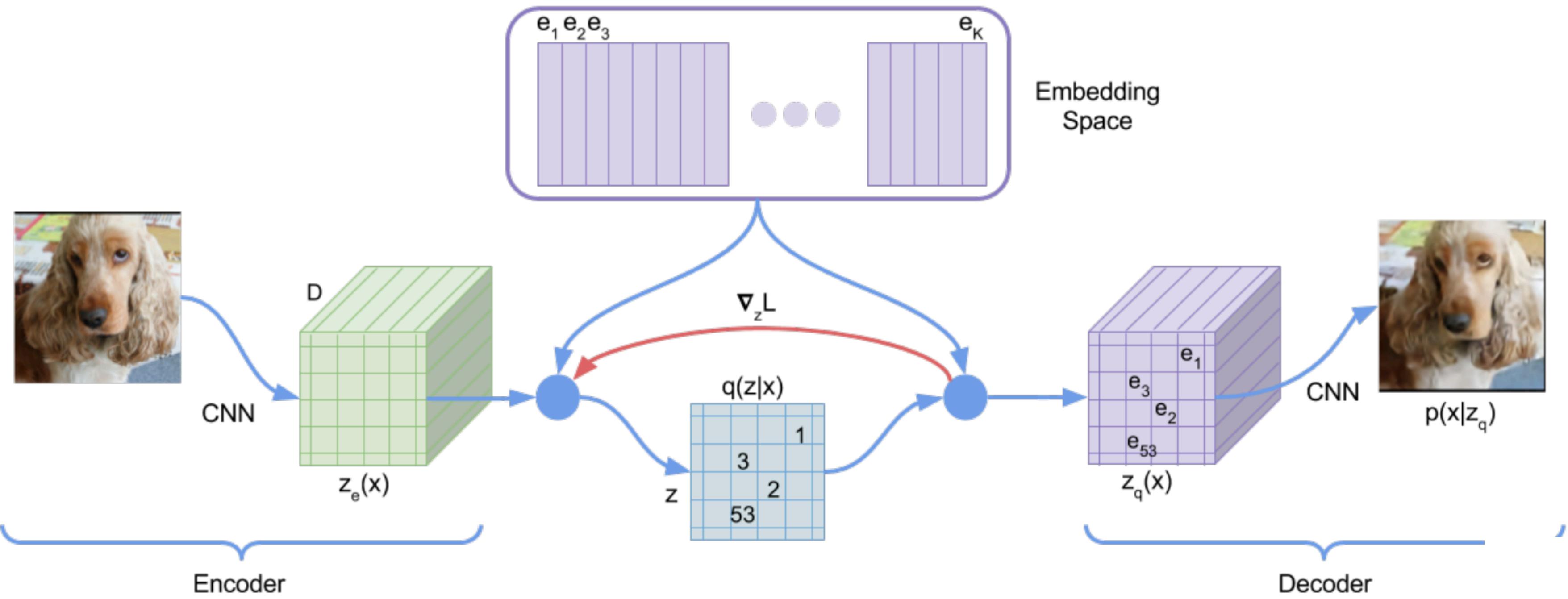
one audio sample at a time

hard to learn long-range dependencies



Audio Codecs: VQ-VAE

Vector quantisation



$$\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{codebook}} + \beta \mathcal{L}_{\text{commit}}$$

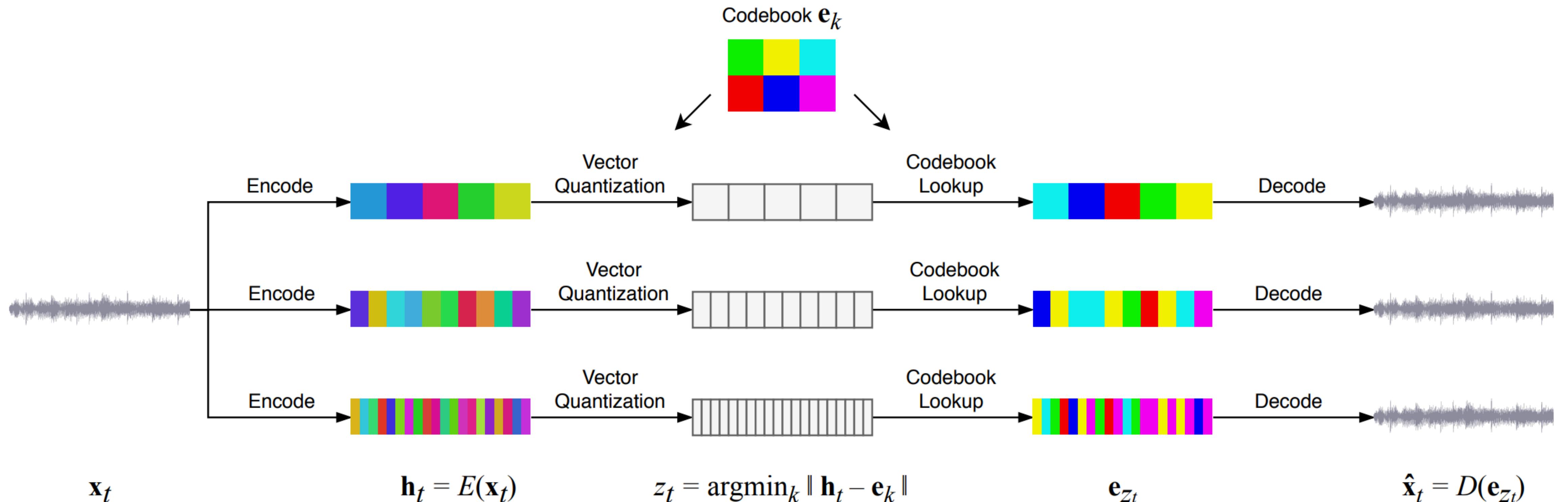
$$\mathcal{L}_{\text{recons}} = \frac{1}{T} \sum_t \|\mathbf{x}_t - D(\mathbf{e}_{z_t})\|_2^2$$

$$\mathcal{L}_{\text{codebook}} = \frac{1}{S} \sum_s \|\text{sg}[\mathbf{h}_s] - \mathbf{e}_{z_s}\|_2^2$$

$$\mathcal{L}_{\text{commit}} = \frac{1}{S} \sum_s \|\mathbf{h}_s - \text{sg}[\mathbf{e}_{z_s}]\|_2^2$$

JukeBox

Multi-level VQ-VAE



JukeBox

Language Model

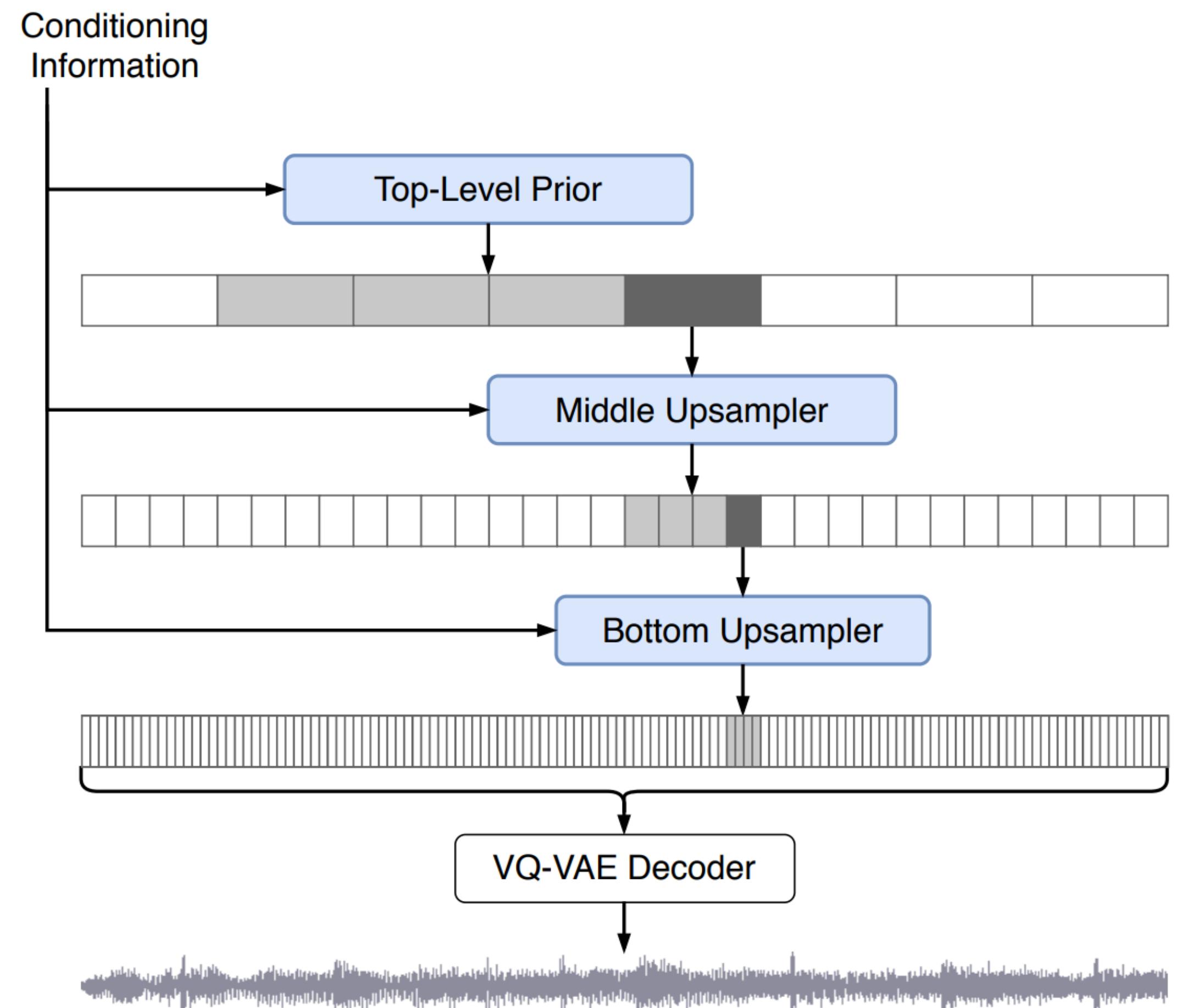
$$\begin{aligned} p(\mathbf{z}) &= p(\mathbf{z}^{\text{top}}, \mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{bottom}}) \\ &= p(\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{middle}}|\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{bottom}}|\mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{top}}) \end{aligned}$$

Details:

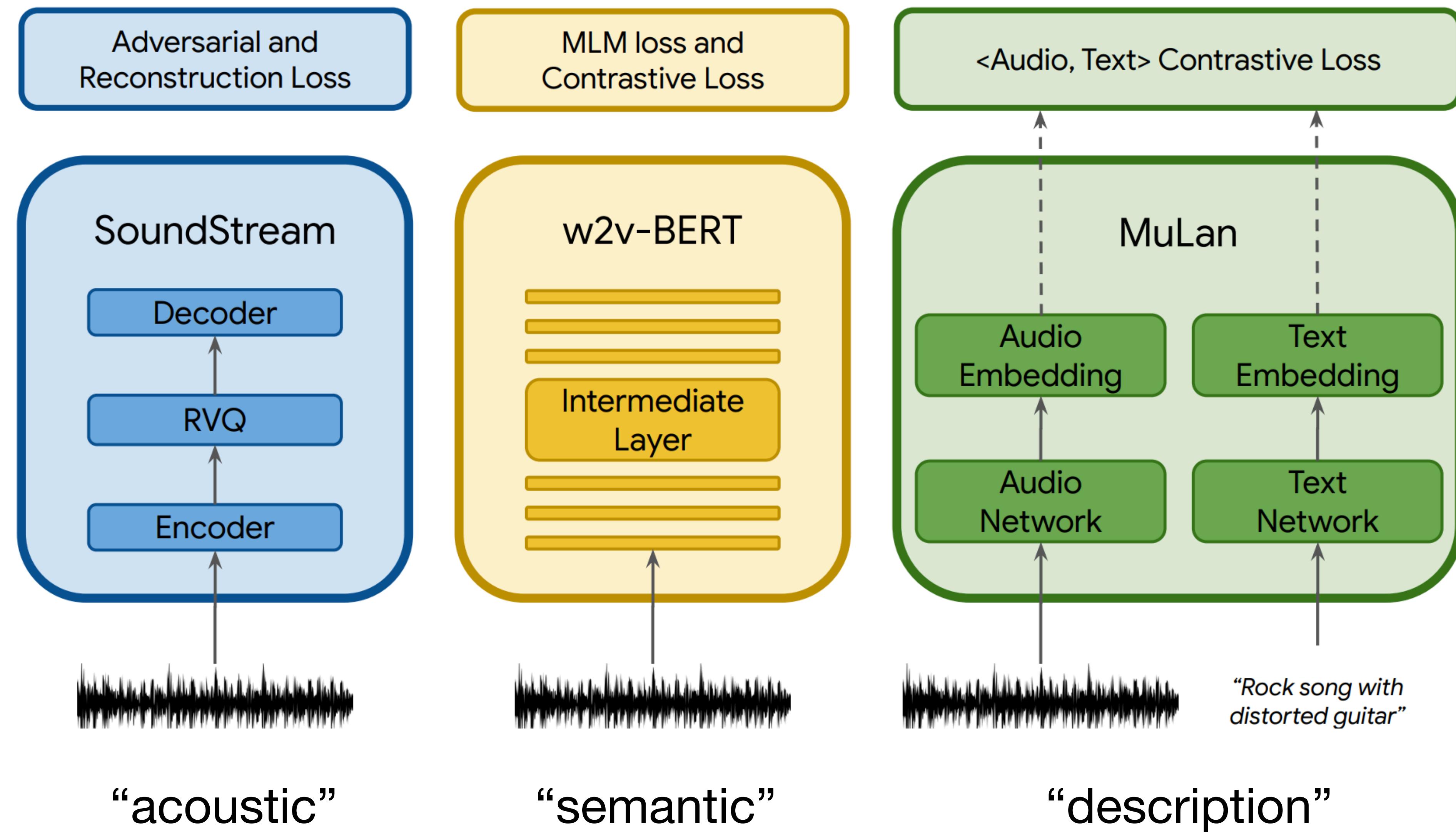
VQ-VAE: 2M parameters, trained on 256 GPU
for 3 days

Upsamplers: 1B parameters,

Top-level: 5B parameters

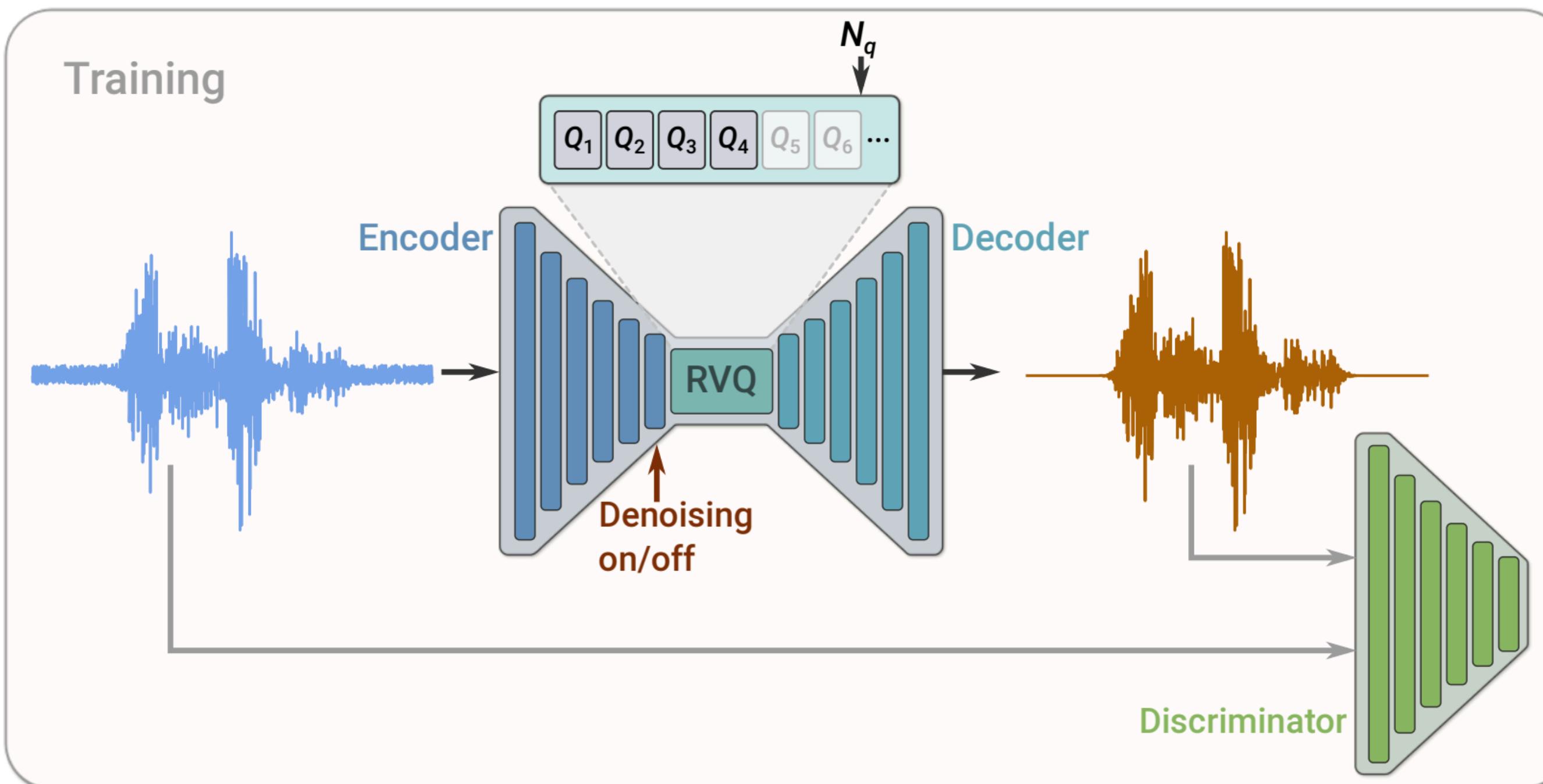


MusicLM (Google)



MusicLM

Acoustic Model (codec): SoundStream



Algorithm 1: Residual Vector Quantization

Input: $y = \text{enc}(x)$ the output of the encoder, vector quantizers Q_i for $i = 1..N_q$

Output: the quantized \hat{y}

```
 $\hat{y} \leftarrow 0.0$ 
 $\text{residual} \leftarrow y$ 
 $\text{for } i = 1 \text{ to } N_q \text{ do}$ 
     $\hat{y} += Q_i(\text{residual})$ 
     $\text{residual} -= Q_i(\text{residual})$ 
 $\text{return } \hat{y}$ 
```

MusicLM

Semantic Model: w2v-BERT

Goal:

raw audio waveforms → latent speech representations

Architecture:

wav2vec 2.0: Extracts context representations from raw waveforms.

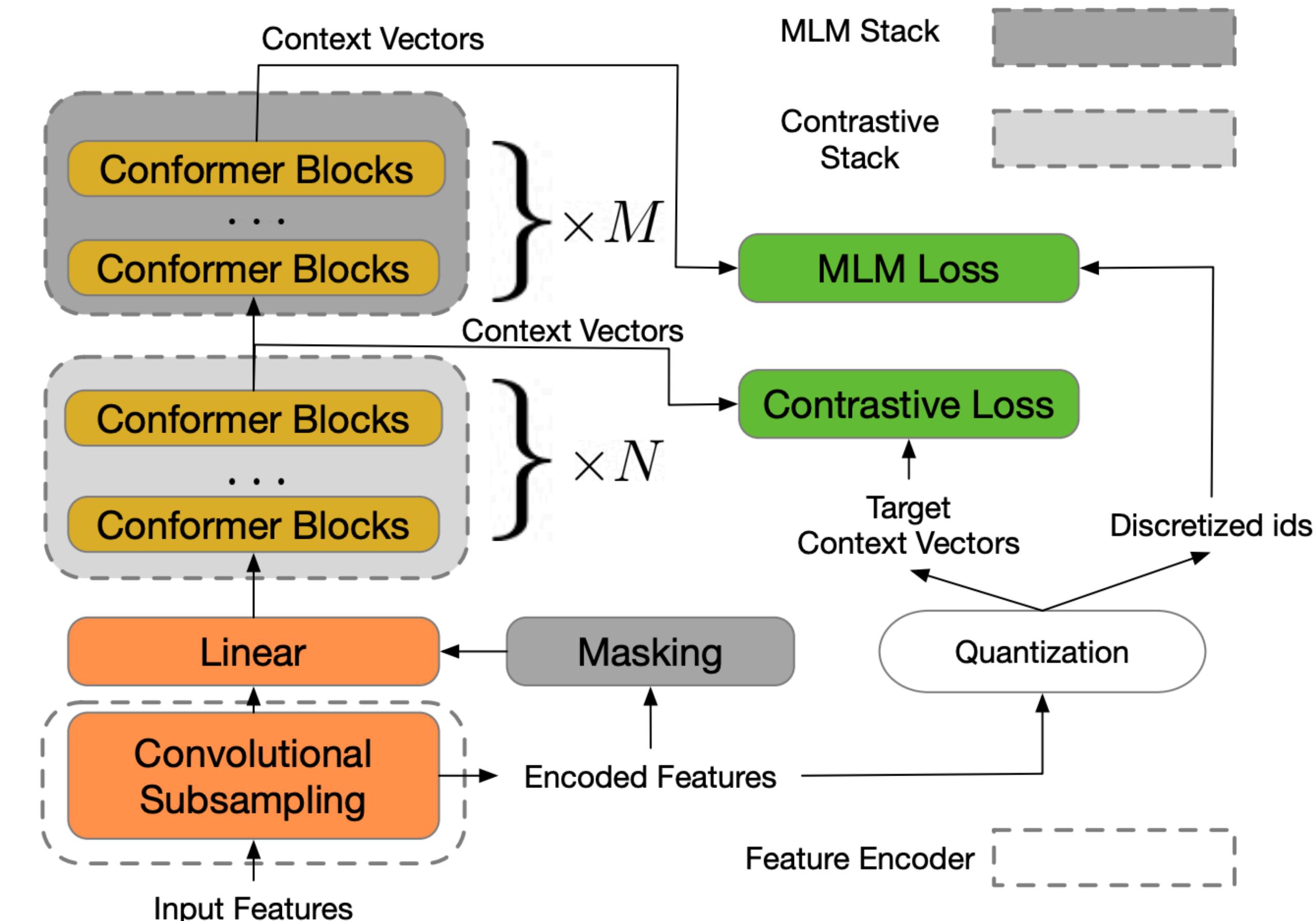
BERT: Models sequence-level dependencies on extracted features.

Objective: Predict masked speech units (like MLM in NLP).

Advantages:

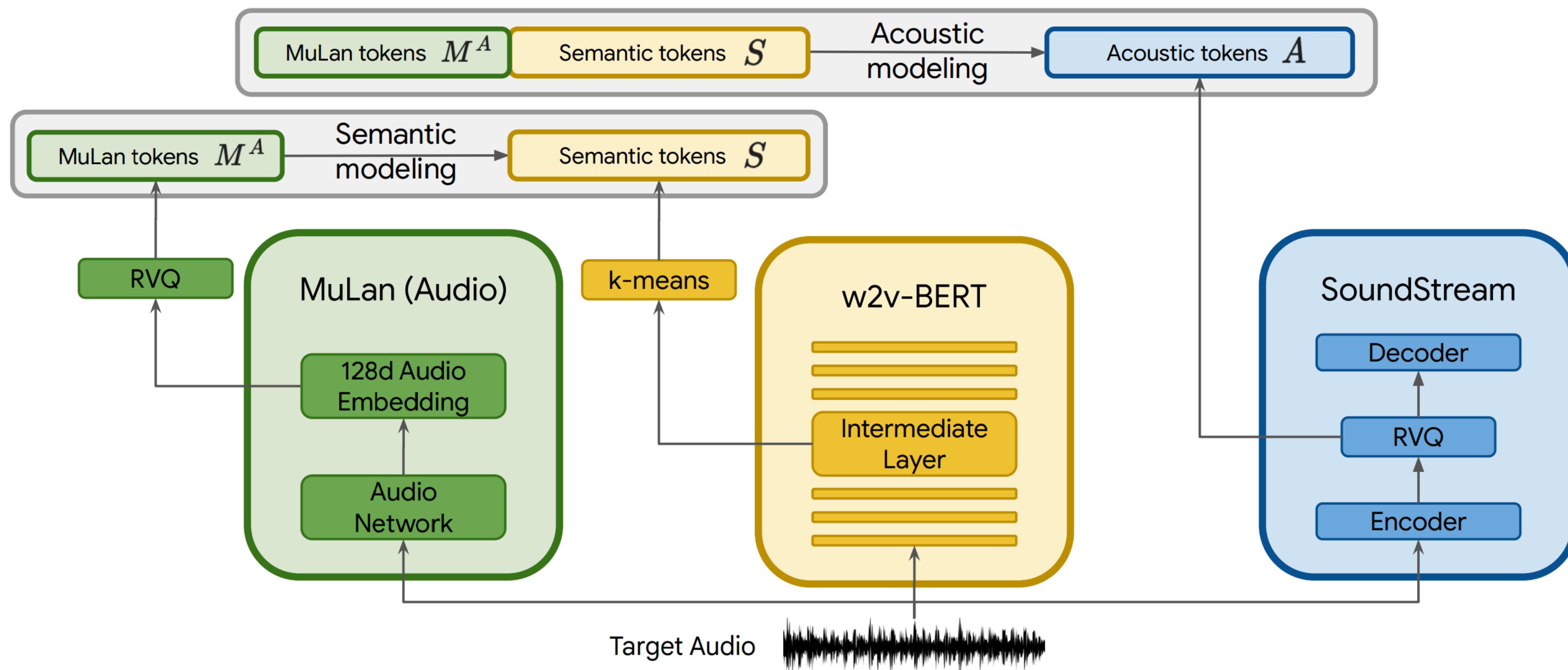
Learns robust speech representations.

Transferable to downstream tasks (ASR, speaker ID, etc.).



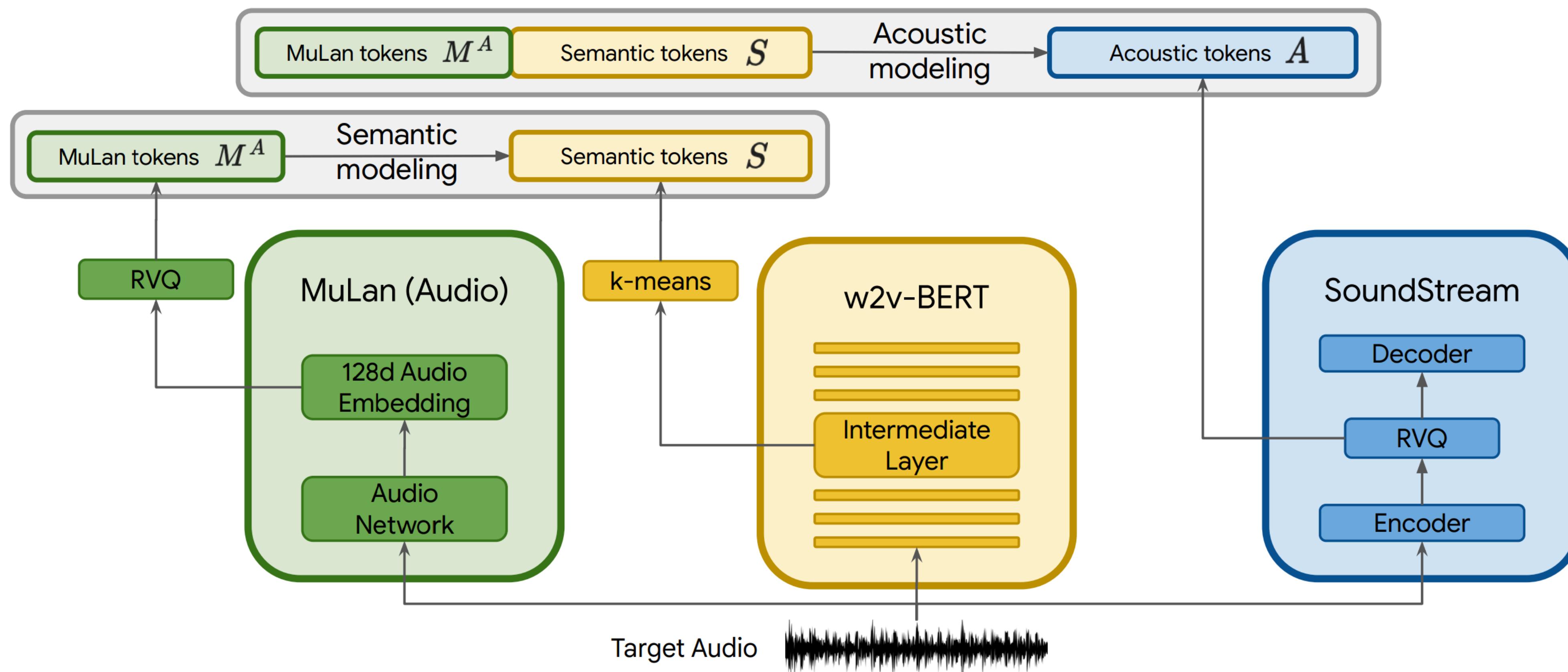
MusicLM

Training



MusicLM

Training

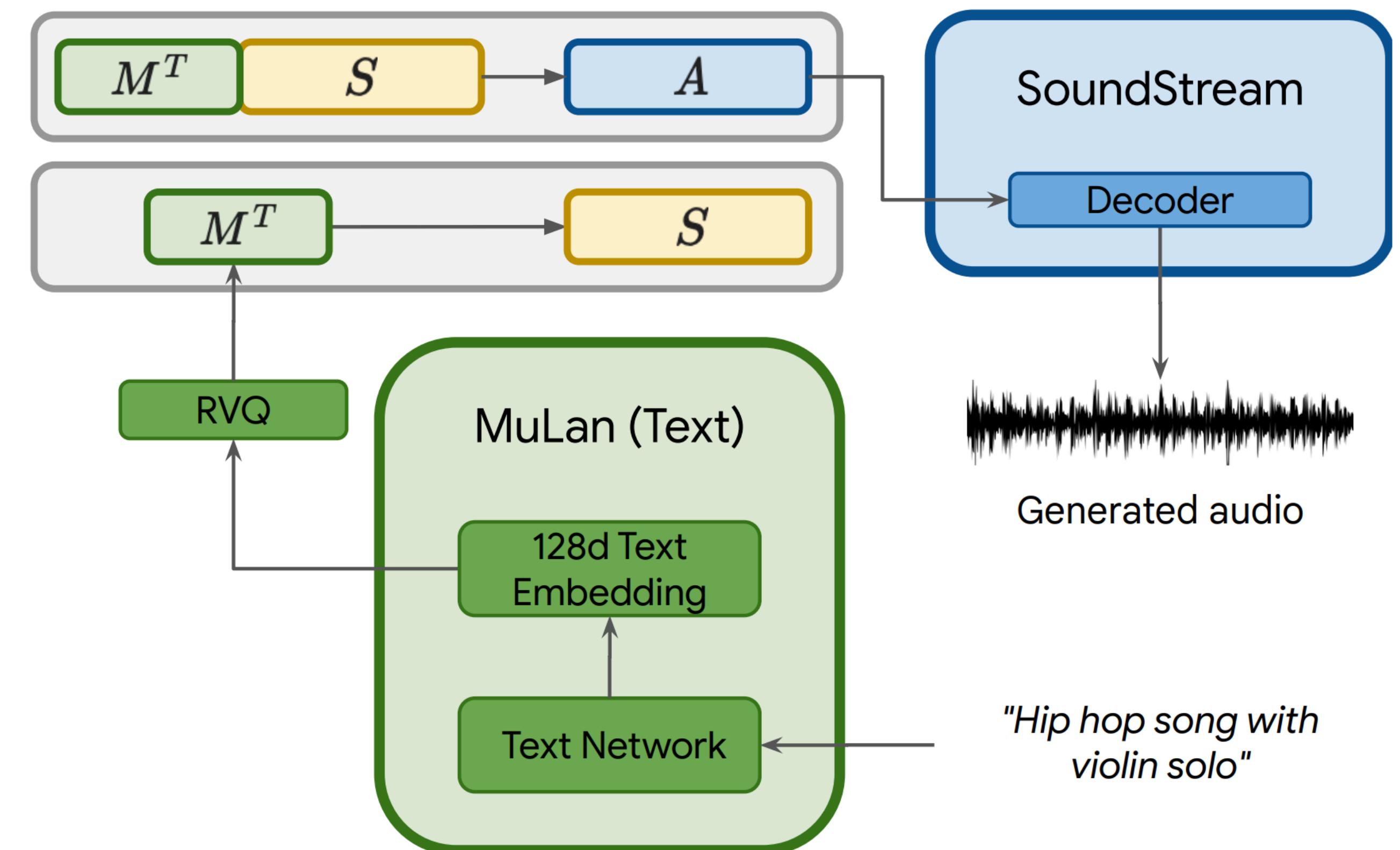


no text descriptions used in training!

MusicLM

Inference

- Input: text description.
- Generation:
 - Text → description tokens.
 - Description tokens → semantic tokens
 - Semantic+description → acoustic
- Decoding: Audio tokens → Full audio using SoundStream.



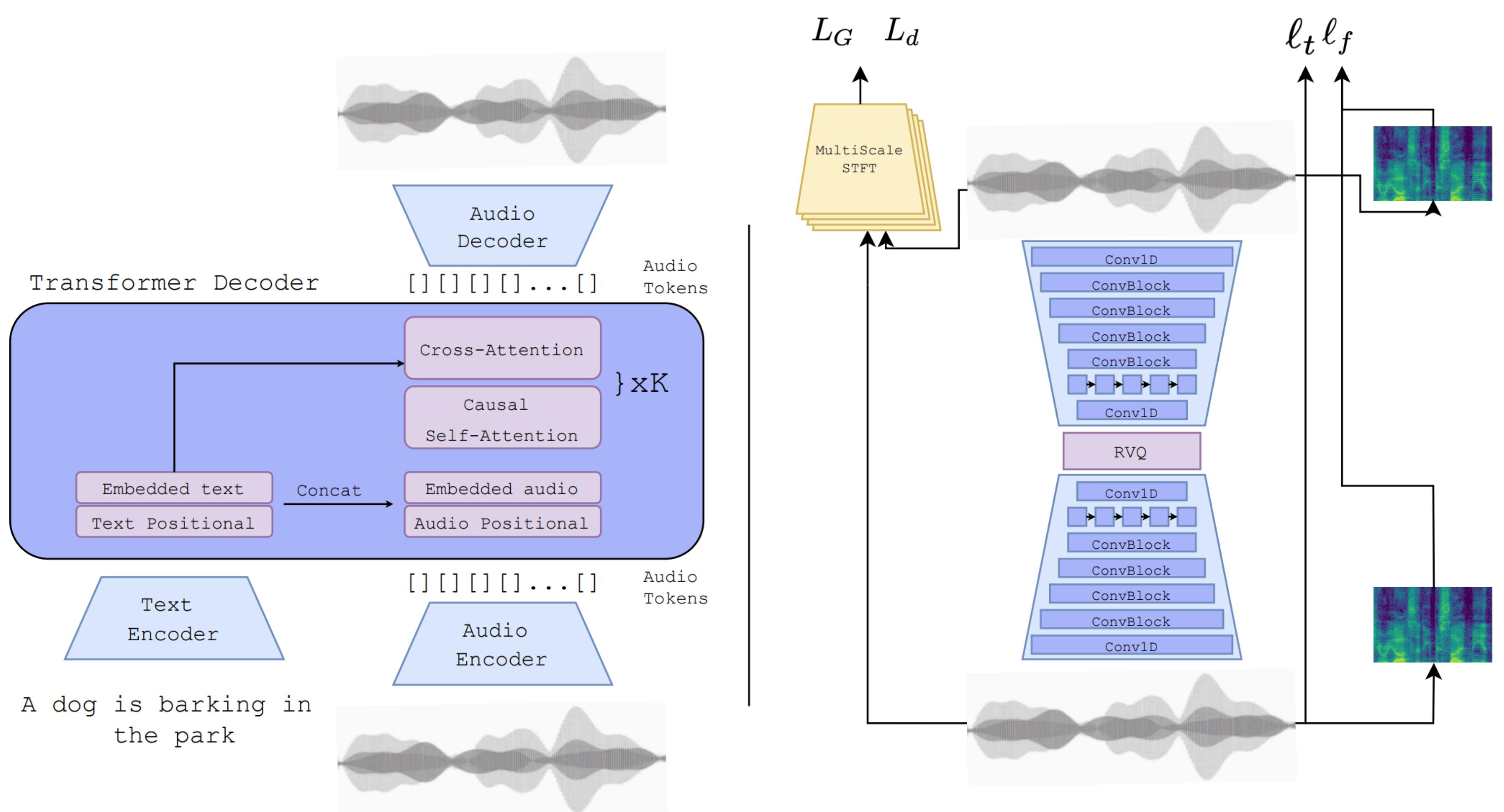
MusicLM overview

MODEL	FAD _{TRILL} ↓	FAD _{VGG} ↓	KLD ↓	MCC ↑	WINS ↑
RIFFUSION	0.76	13.4	1.19	0.34	158
MUBERT	0.45	9.6	1.58	0.32	97
MUSICLM	0.44	4.0	1.01	0.51	312
MUSICCAPS	-	-	-	-	472

- Training dataset: 280k hours
- All components are proprietary.
- Open source analogues:
 - MuLan: CLAP
 - SoundStream: EnCodec
 - w2v-BERT: MERT or MusicFM

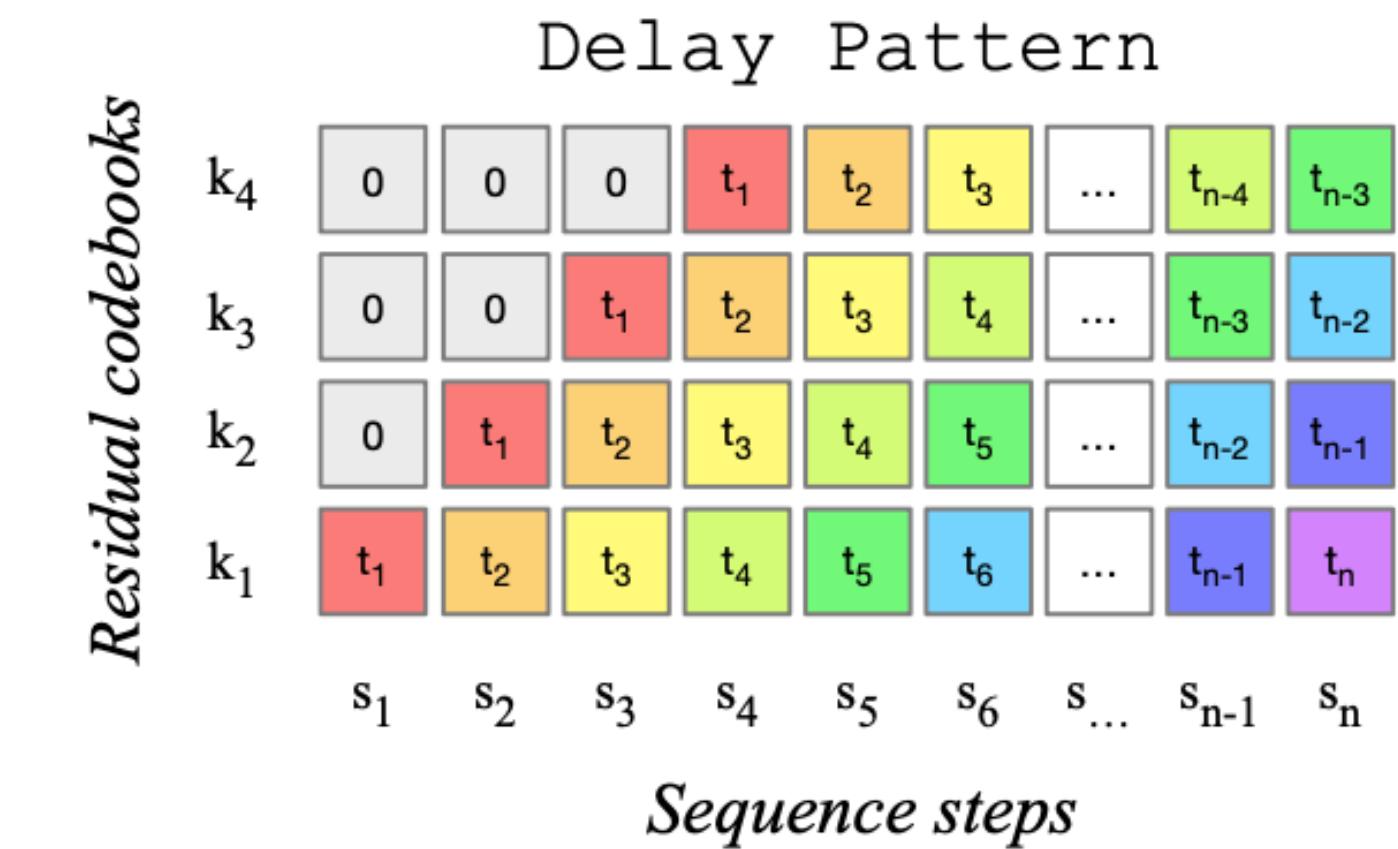
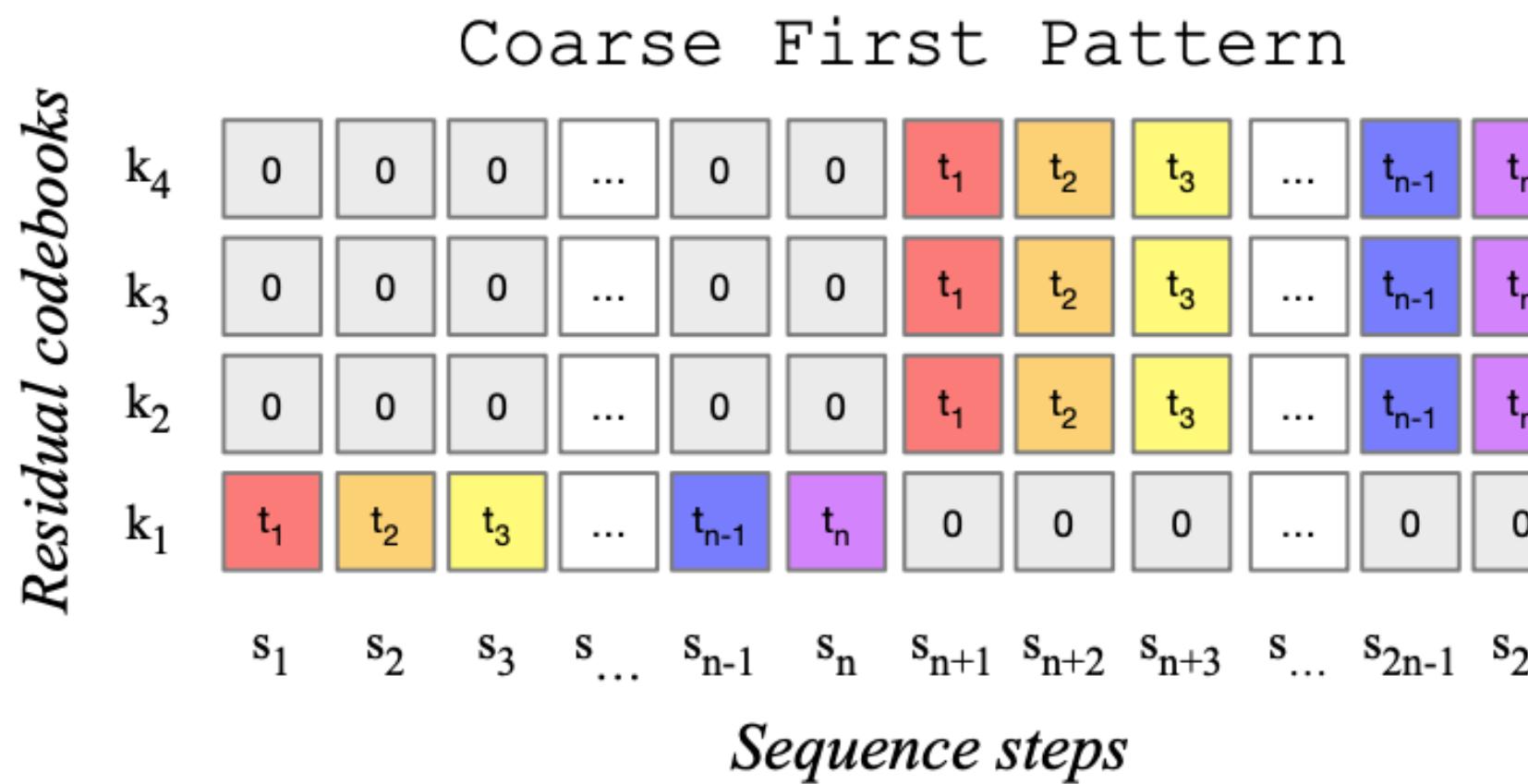
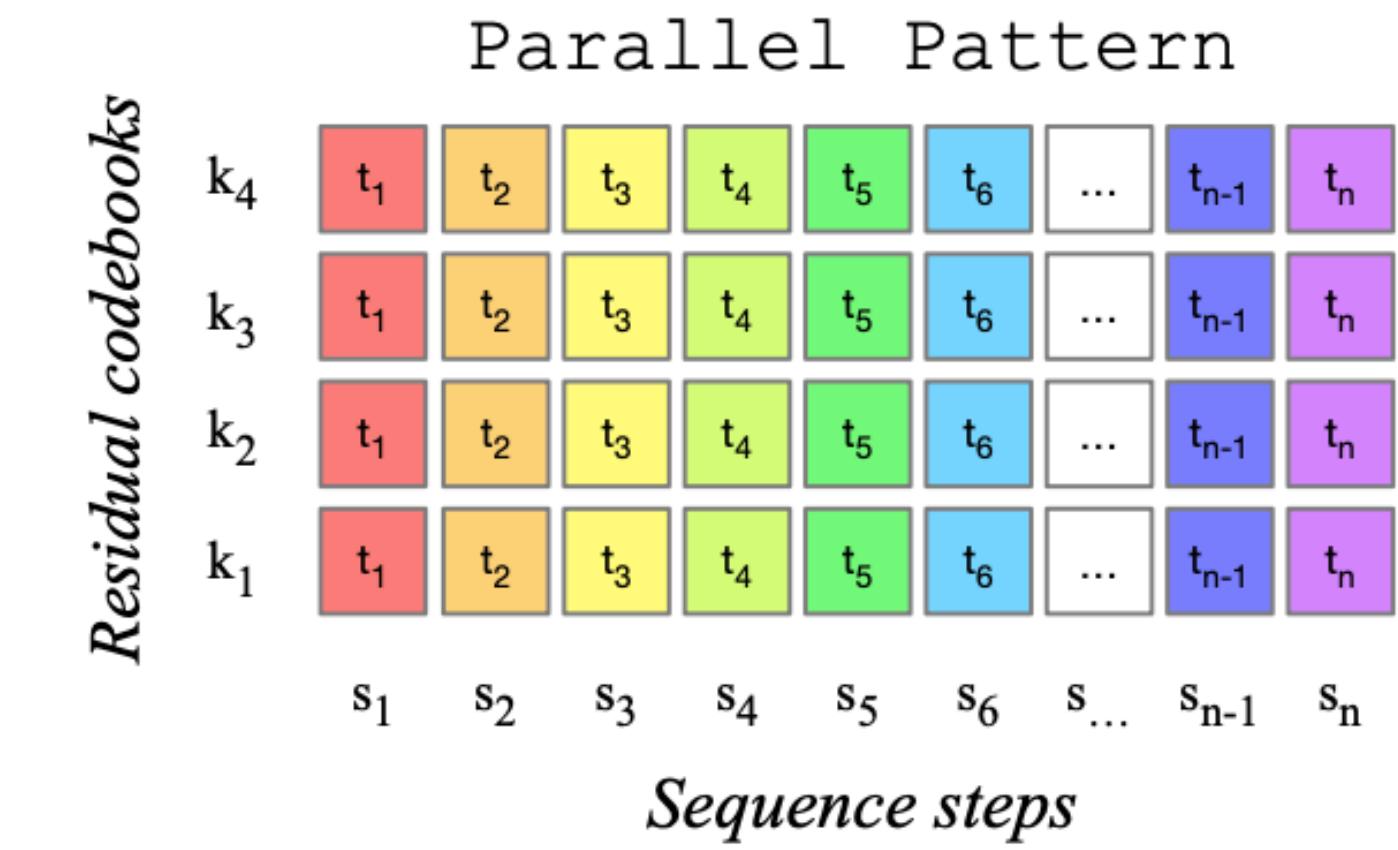
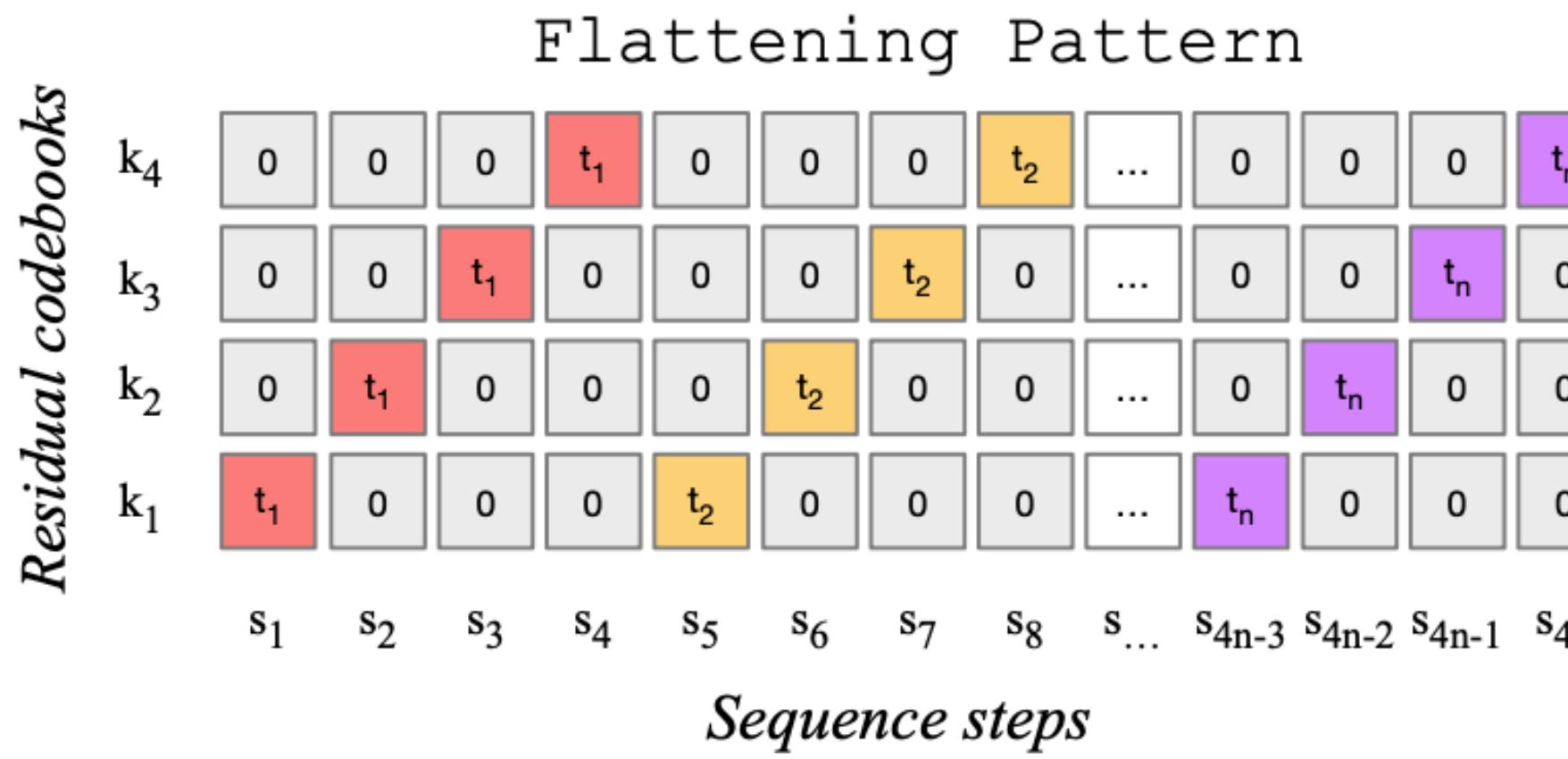
MusicGEN

EnCodec



MusicGEN

Flattening/Interleaving



MusicGEN

overview

Audio Encoding: RVQGAN (EnCodec)

Generative Model: Conditioned Autoregressive Transformer

Text conditioning: prefix + cross-attention

5-layers EnCodec, 50Hz, 4 RVQ quantisers, codebook size 2048

300M - 3.3B parameters in Transformer (trained on 32-96 GPUs)

Training data: 20k hours

MusicGEN

Reported results

MODEL	MUSICCAPS Test Set				
	FAD _{vgg} ↓	KL ↓	CLAP _{scr} ↑	OVL. ↑	REL. ↑
Riffusion	14.8	2.06	0.19	79.31±1.37	74.20±2.17
Mousai	7.5	1.59	0.23	76.11±1.56	77.35±1.72
MusicLM	4.0	-	-	80.51±1.07	82.35±1.36
Noise2Music	2.1	-	-	-	-
MUSICGEN w.o melody (300M)	3.1	1.28	0.31	78.43±1.30	81.11±1.31
MUSICGEN w.o melody (1.5B)	3.4	1.23	0.32	80.74±1.17	83.70±1.21
MUSICGEN w.o melody (3.3B)	3.8	1.22	0.31	84.81±0.95	82.47±1.25
MUSICGEN w. random melody (1.5B)	5.0	1.31	0.28	81.30±1.29	81.98±1.79

Questions