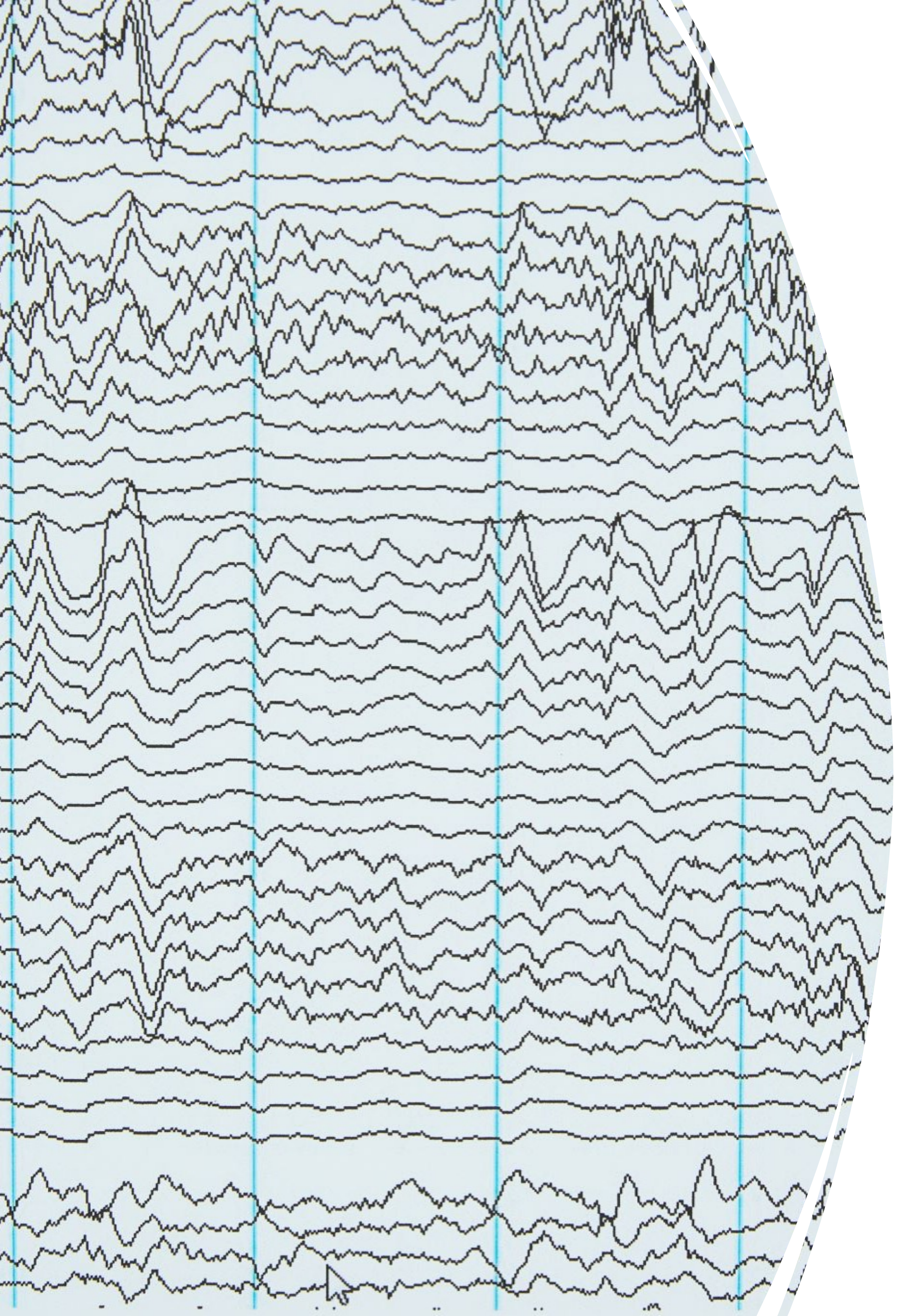


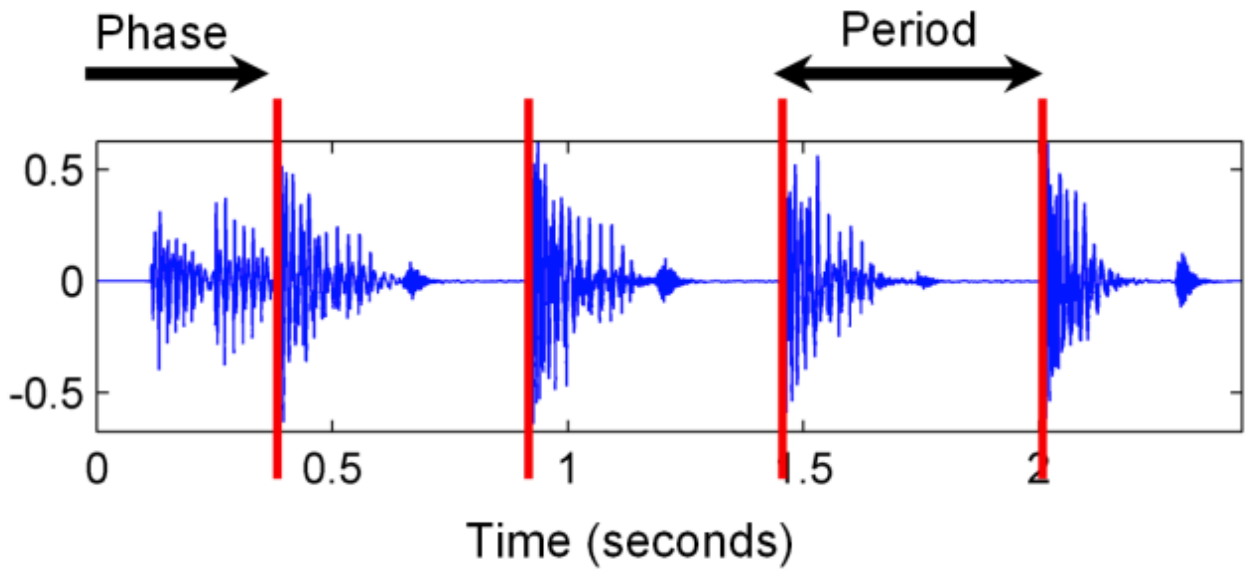
Temporal convolutional networks for musical audio beat tracking – Davies & Bock (2019)

Matthew Rice
220503466

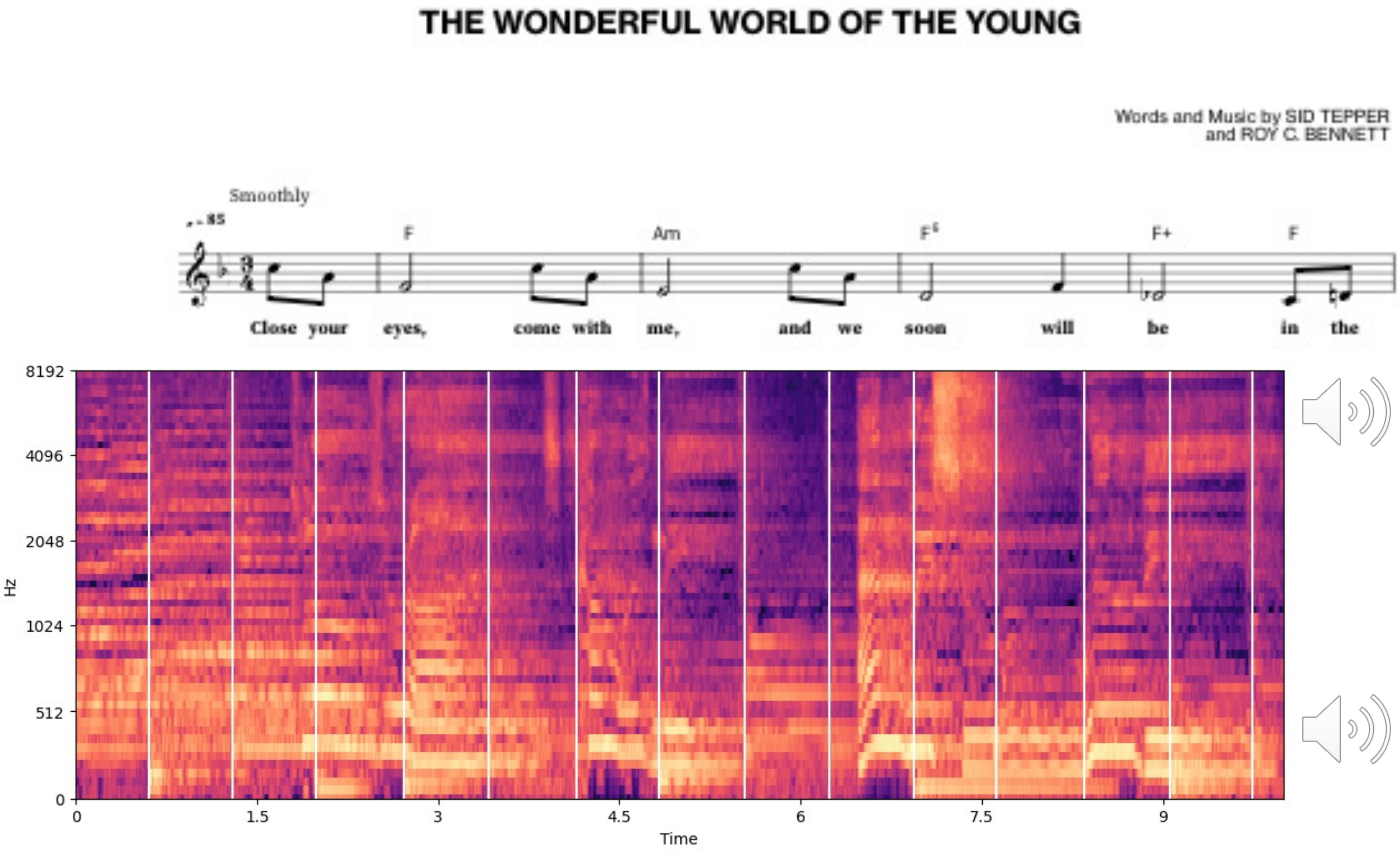
What is Beat Tracking?

From S Dixon slides on Rhythm and Metre:

- Pulse – an equally spaced sequence of **perceived** accents in time
- Primary pulse(beat/tactus) – rate at which one taps along with music
- Constant tempo/timing not assumed



Example

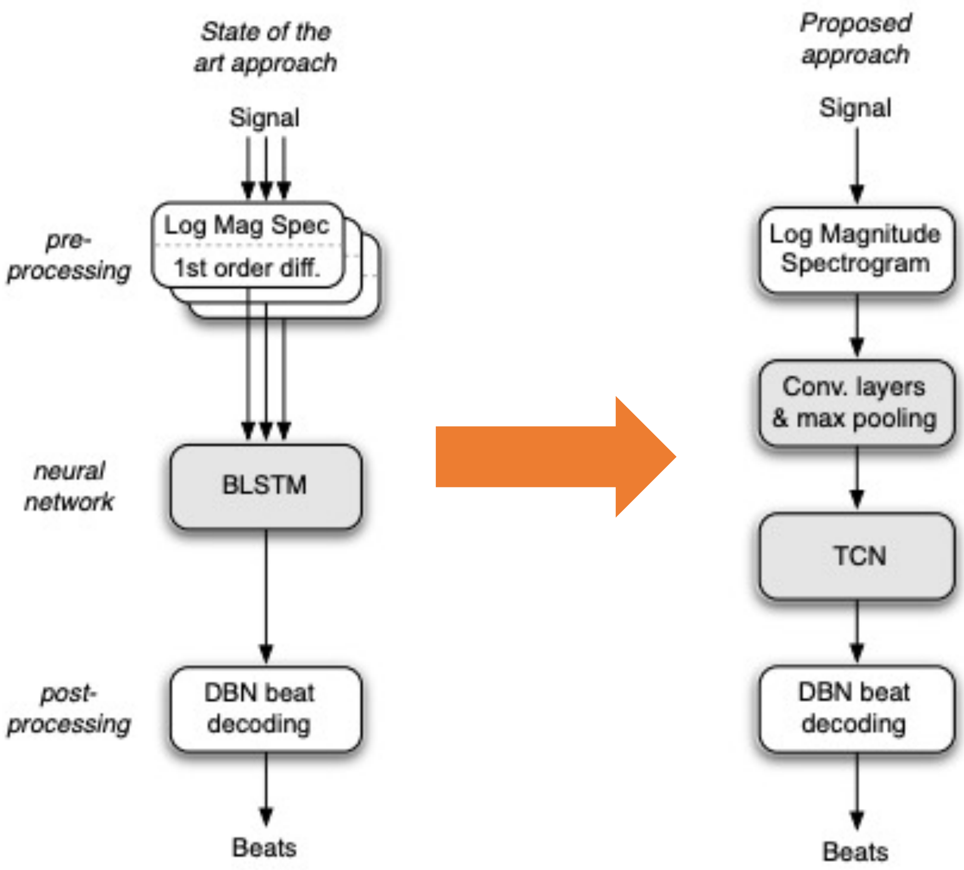


THE WONDERFUL WORLD OF THE YOUNG
Words and Music by SID TEPPER
and ROY C. BENNETT
Smoothly
F
Am
F+
F
Close your eyes, come with me, and we soon will be in the
8192
4096
2048
1024
512
0
Hz
0
1.5
3
4.5
6
7.5
9
Time

Previous Work

- Beat Tracking by Dynamic Programming (Daniel P. W. Ellis 2007)
  - Match Onset Strength to estimated global tempo
- An Efficient State-Space Model for Joint Tempo and Meter Tracking (Krebs et al. 2015)
  - Efficient implementation of Dynamic Bayesian Network (**DBN**) for improved accuracy (> 15%)
- Joint Beat and Downbeat Tracking with RNNs (Bock et al. 2016)
  - Use RNN to predict musical onsets combined with DBN to find global best state sequence
  - Limitations: RNNs hard to train, uninterpretable, and inefficient

Main Idea: Use TCN instead of RNN



TCN + Conv. Layers

+ Efficiency
+ Interpretability
+ GPU parallelizable
◯ Similar performance to RNN

WaveNet
(Oord et al. 2016)

Introduction of Temporal Convolutional Network (TCN) – Uses stacked dilated causal convolutions

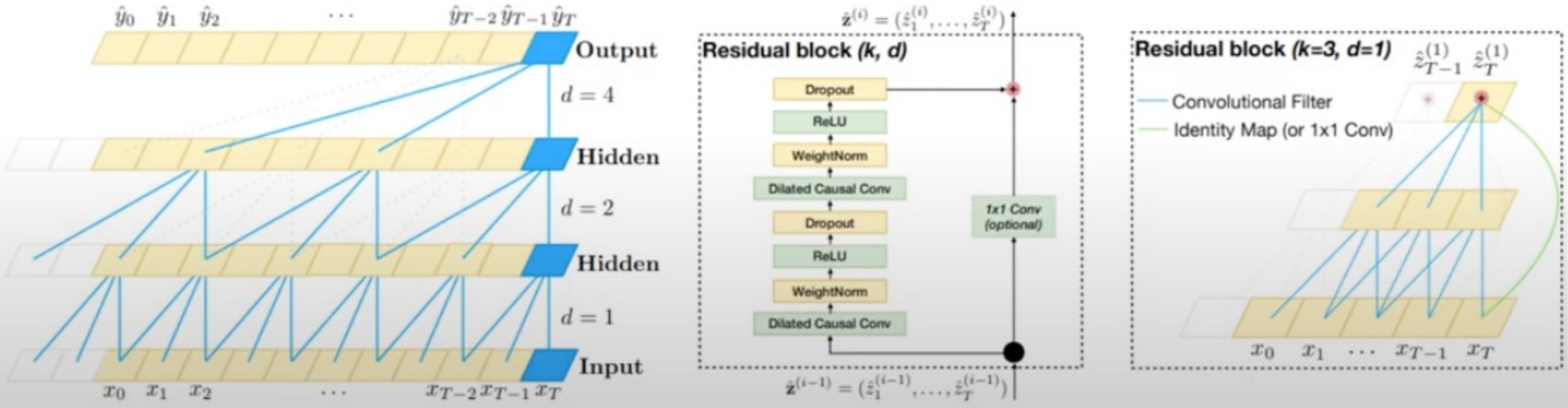


Image from VU Amsterdam: *Lecture 5.4 - CNNs for Sequential Data*, (Nov. 04, 2020).

Datasets

TABLE II
OVERVIEW OF THE DATASETS USED FOR TRAINING AND EVALUATION.

Dataset	# files	length
Ballroom [22], [23] [1]	685	5 h 57 m
Beatles [19]	180	8 h 09 m
Hainsworth [24]	222	3 h 19 m
Simac [25]	595	3 h 18 m
SMC [26]	217	2 h 25 m
GTZAN [20], [21]	999	8 h 20 m

8-fold cross validation

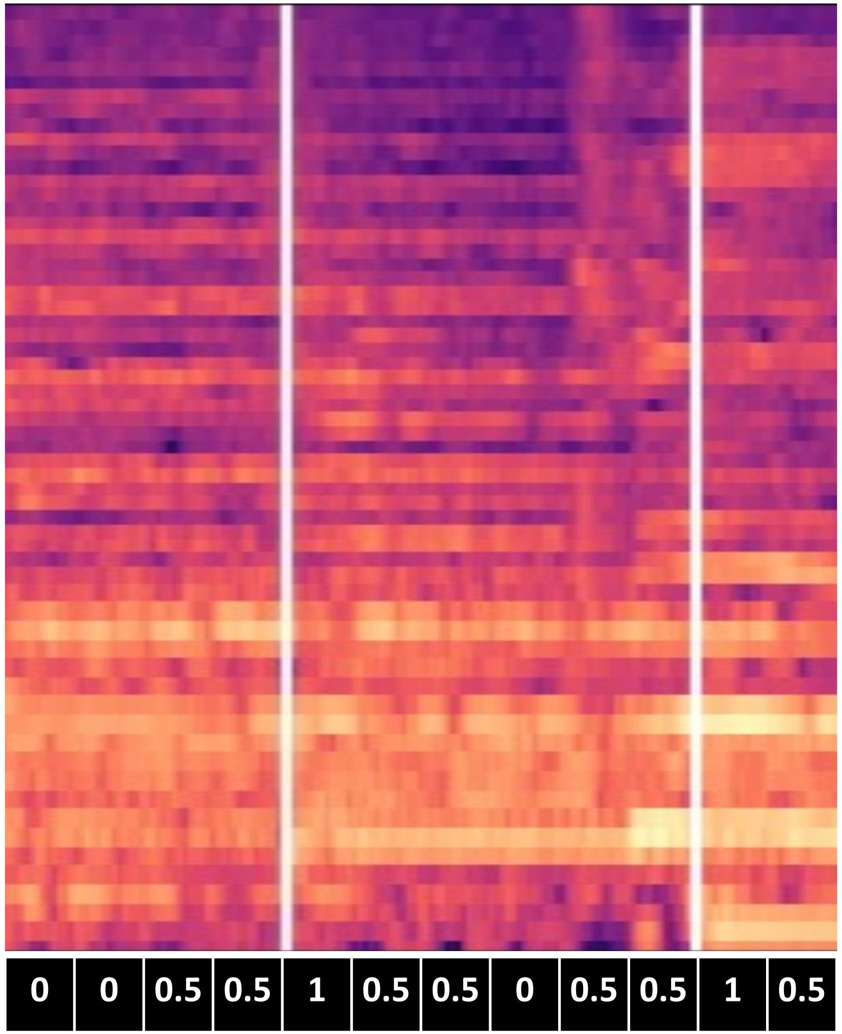
Held out test set

Method

Formulation: Treat problem as binary classification task

Input data format

- Song:
  - Spectrogram: (batch, bands, time windows)
- Labels
  - (batch, time windows)
  - 0 → no beat
  - 1 → beat
  - 0.5 → 2 locations adjacent to beat



Method

3 Layer CNN to reduce 5 frames (5x81) to (1x16)

Unlike WaveNet:

- Non-causal convolutions
- ELU activation

Same as in RNN paper
Only used during inference

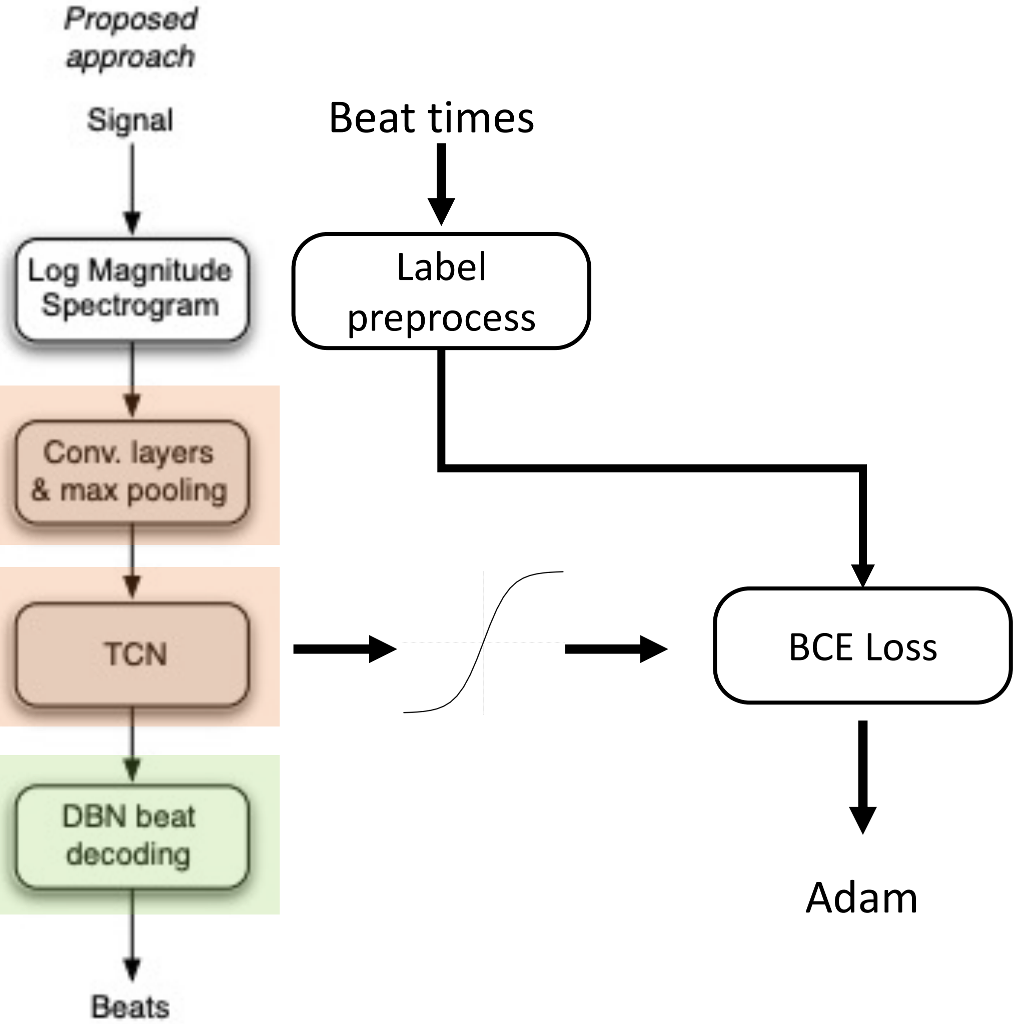


TABLE I
OVERVIEW OF SIGNAL PROCESSING AND LEARNING PARAMETERS

Signal Conditioning	
Audio sample rate	44.1 kHz
Window shape	*Hann*
Window & FFT size	2048 *samples*
Hop size	10 ms
Filterbank freq. range	30 . . . 17000 Hz
Sub-bands per octave	12
Total number of bands	81
Conv. Block	
Number of filters	16, 16, 16
Filter size	3×3, 3×3, 1×8
Max. pooling size	1×3, 1×3, —
Dropout rate	0.1
Activation function	*ELU*
TCN	
Number of stacks	1
Dilations	$2^{0,\ldots,10}$
Number of filters	16
Filter size	5
Spatial dropout rate	0.1
Activation function	*ELU*
Training	
Optimizer	*Adam*
Learning rate	0.001
Batch size	1
Output activation function	*sigmoid*
Loss function	*binary cross-entropy*

Results

TABLE III
OVERVIEW OF BEAT TRACKING PERFORMANCE.

	F-measure	CMLc	CMLt	AMLc	AMLt	D
Ballroom						
TCN	0.933	0.864	0.881	0.909	0.929	3.456
BLSTM [5]	0.917	0.832	0.849	0.905	0.926	**3.539**
BLSTM [6]	**0.938**	**0.872**	**0.892**	**0.932**	**0.953**	3.397
Hainsworth						
TCN	0.874	0.755	0.795	**0.882**	**0.930**	**3.518**
BLSTM [5]	**0.884**	**0.769**	**0.808**	0.873	0.916	3.507
BLSTM [6]	0.871	0.732	0.784	0.849	0.910	3.395
SMC						
TCN	**0.543**	**0.315**	**0.432**	**0.462**	**0.632**	**1.574**
BLSTM [5]	0.529	0.296	0.428	0.383	0.567	1.460
BLSTM [6]	0.516	0.307	0.406	0.429	0.575	1.514
GTZAN						
TCN	0.843	0.695	0.715	0.889	0.914	**3.096**
BLSTM [5]	**0.864**	**0.750**	**0.768**	**0.901**	**0.927**	3.071
BLSTM [6]	0.856	0.716	0.744	0.876	0.919	3.019

Limitations

- No joint beat and downbeat tracking
- Large number of parameters not completely tuned
- Inference slower than BLSTM (but still faster than real-time)
- Non-causal TCN layers inhibit real-time tasks

Implications

- TCNs can be adapted from music generation domain to beat tracking (and perhaps other domains!)
- TCNs are efficient compared to RNNs while achieving similar performance
  - 35% of weights of RNN approach
  - 60x training speed up on GPU
- Reusing post-processing ideas such as DBN are essential to improving accuracy