How do we extract audio features?

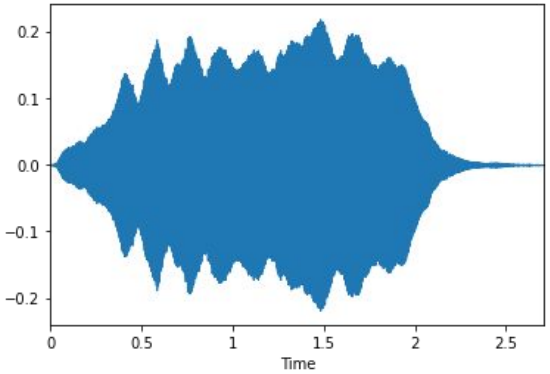
Valerio Velardo

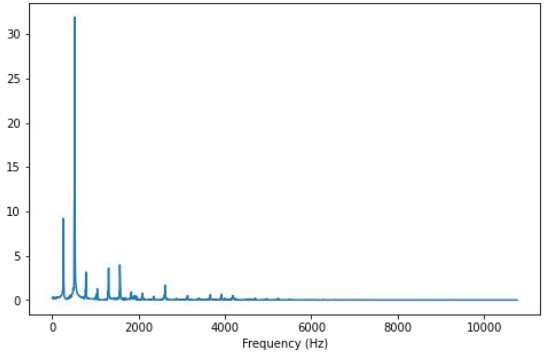
Join the community!

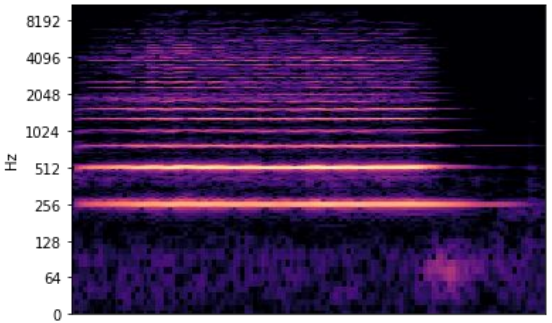
thesoundofai.slack.com

Previously on Audio Processing for ML

- Time-domain features
- Frequency-domain features
- Time-frequency domain features

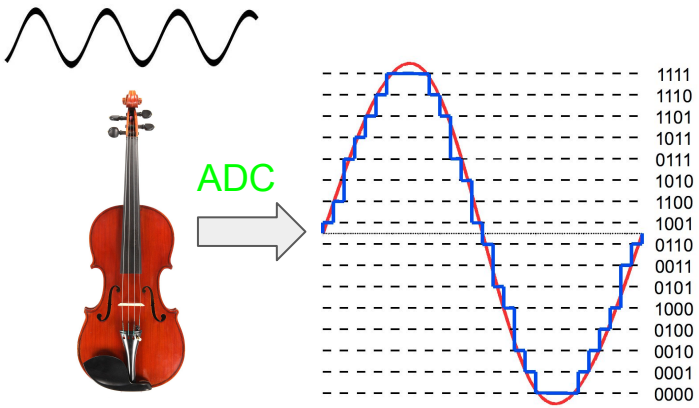




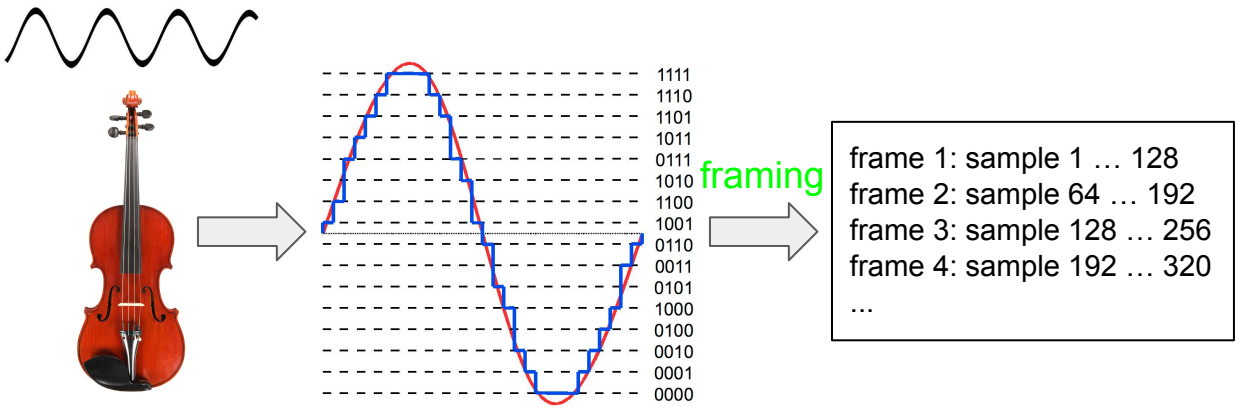


Time-domain feature pipeline

Time-domain feature pipeline



Time-domain feature pipeline



Frames

- Perceivable audio chunk

Frames

- Perceivable audio chunk

1 sample @44.1KHz = 0.0227ms

Frames

- Perceivable audio chunk

1 sample @44.1KHz = 0.0227ms

Duration 1 sample << Ear’s time resolution (10ms)

Frames

- Perceivable audio chunk
- Power of 2 num. samples

Frames

- Perceivable audio chunk
- Power of 2 num. samples
- Typical values: 256 - 8192

Frames

- Perceivable audio chunk
- Power of 2 num. samples
- Typical values: 256 - 8192

$$d_f = \frac{1}{s_r} \cdot K$$

Frames

- Perceivable audio chunk
- Power of 2 num. samples
- Typical values: 256 - 8192

$$d_f = \frac{1}{s_r} \cdot K$$

512

44100

Frames

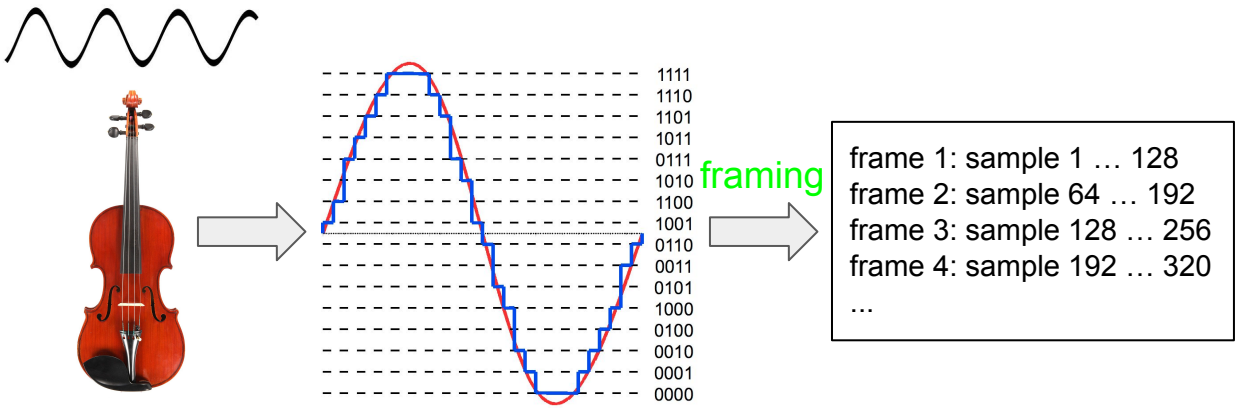
- Perceivable audio chunk
- Power of 2 num. samples
- Typical values: 256 - 8192

$$d_f = \frac{1}{s_r} \cdot K = 11.6\text{ms}$$

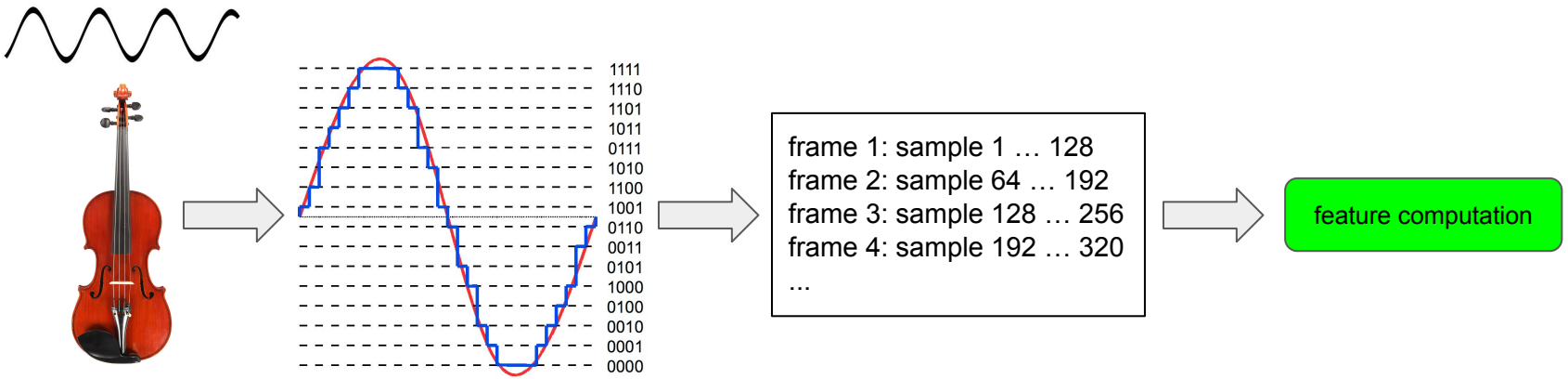
512

44100

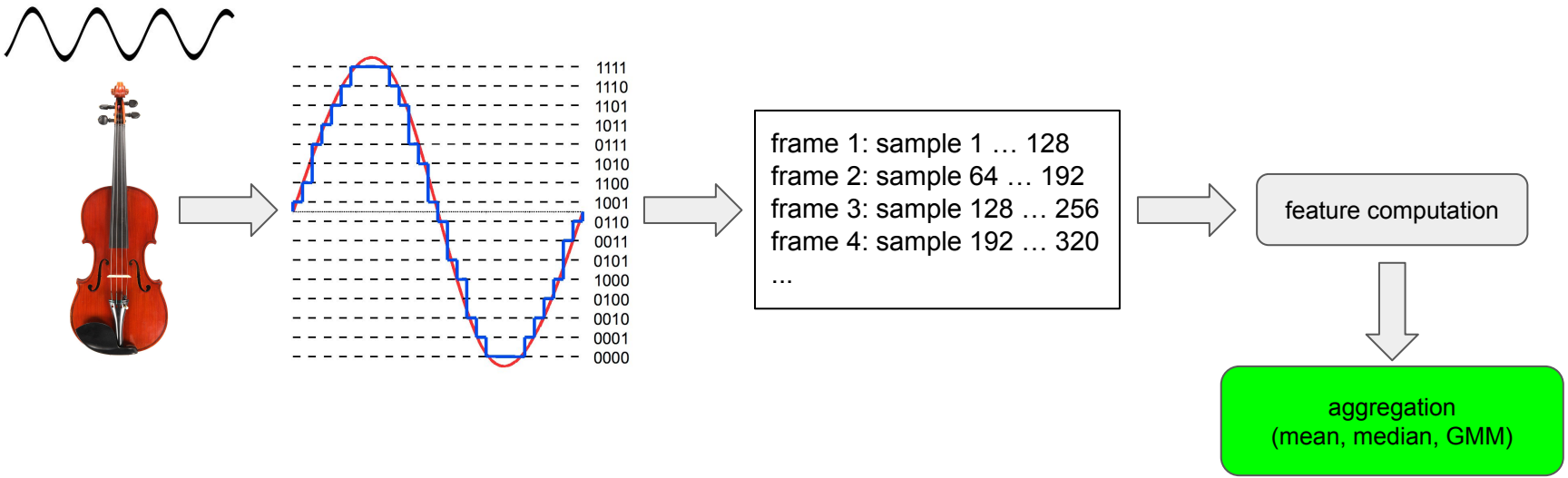
Time-domain feature pipeline



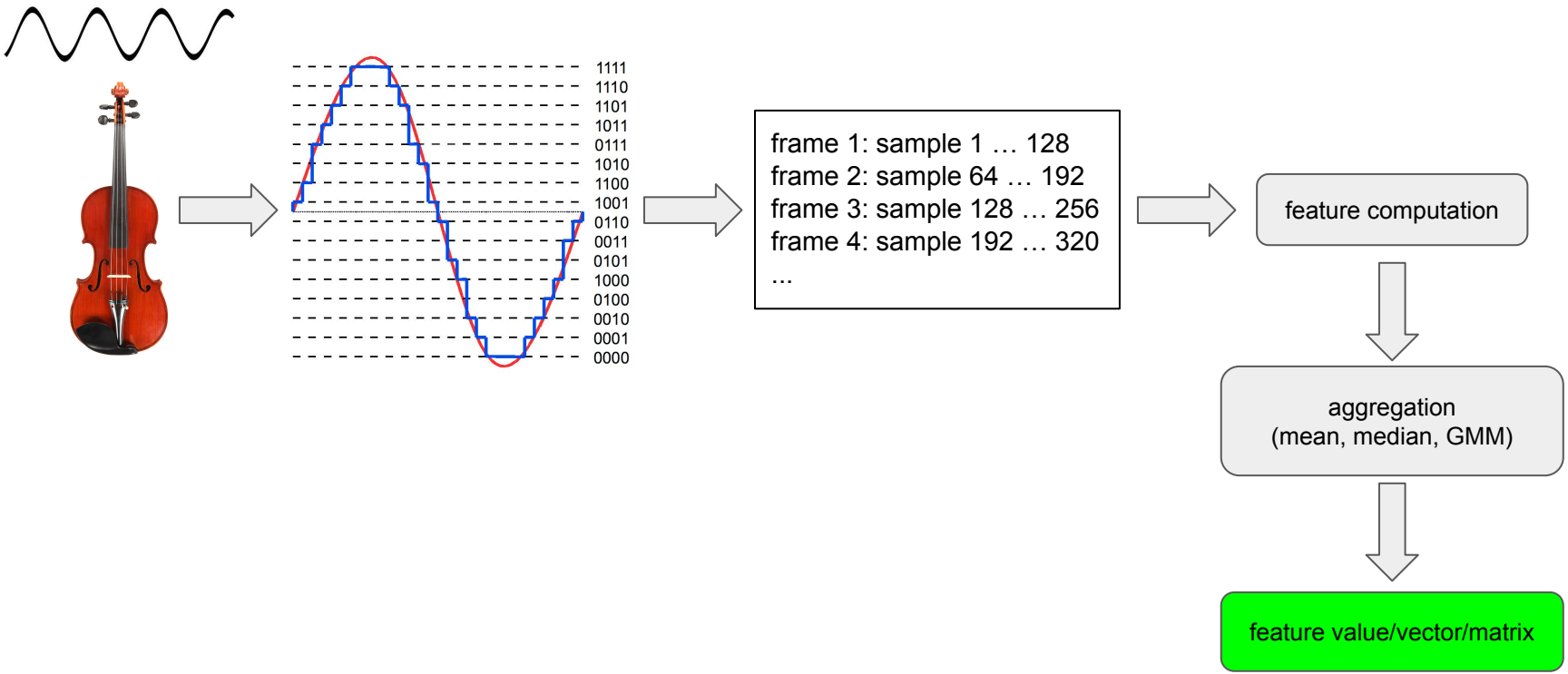
Time-domain feature pipeline



Time-domain feature pipeline

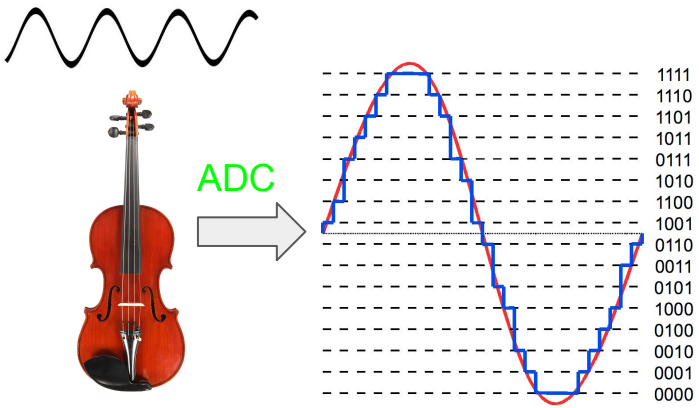


Time-domain feature pipeline

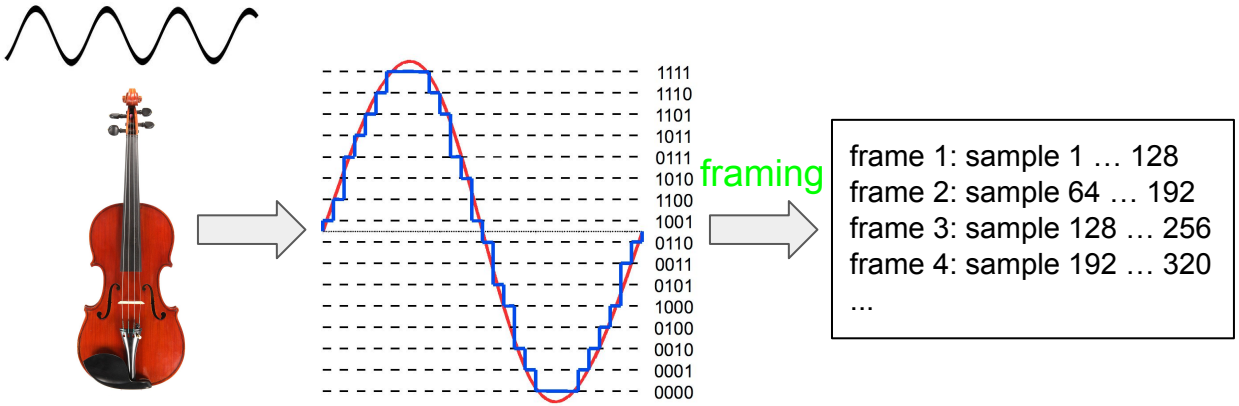


Frequency-domain feature pipeline

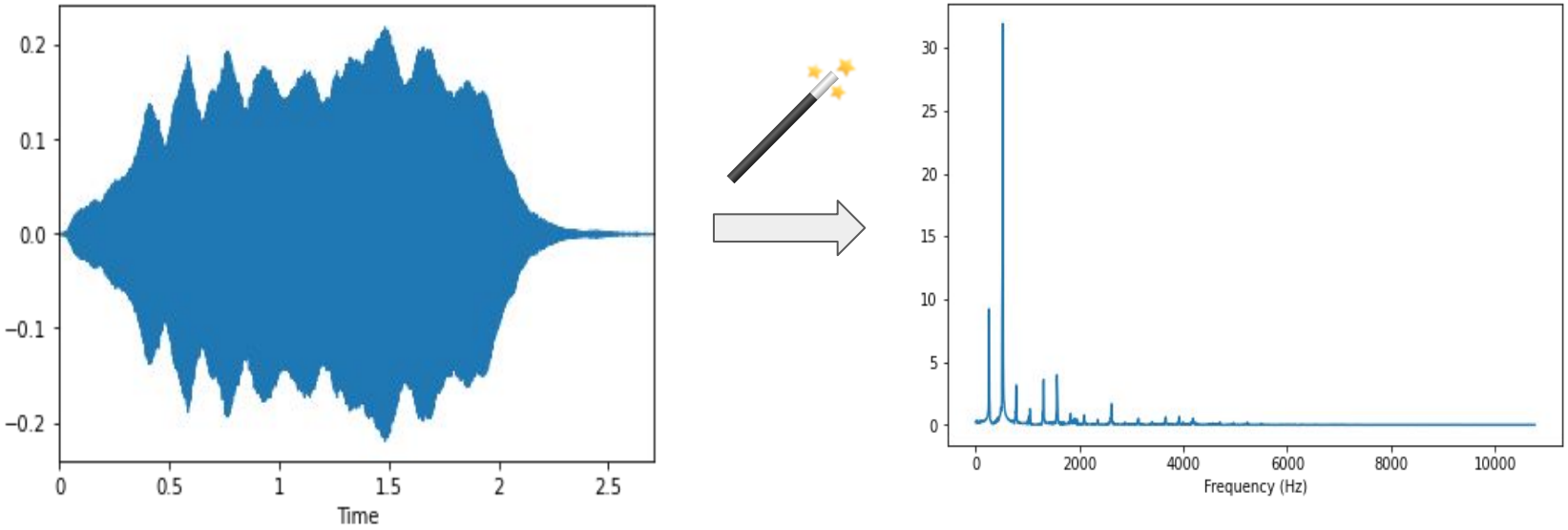
Frequency-domain feature pipeline



Frequency-domain feature pipeline



From time to frequency domain



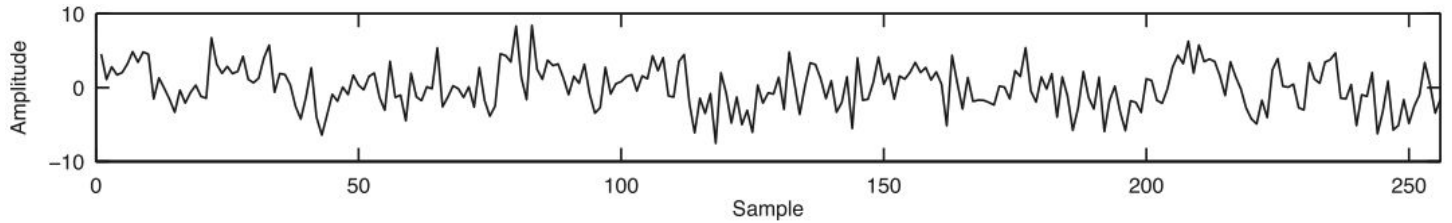

I NEVER THOUGHT OF
SPECTRAL LEAKAGE
imgflip.com

Spectral leakage

- Processed signal isn’t an integer number of periods

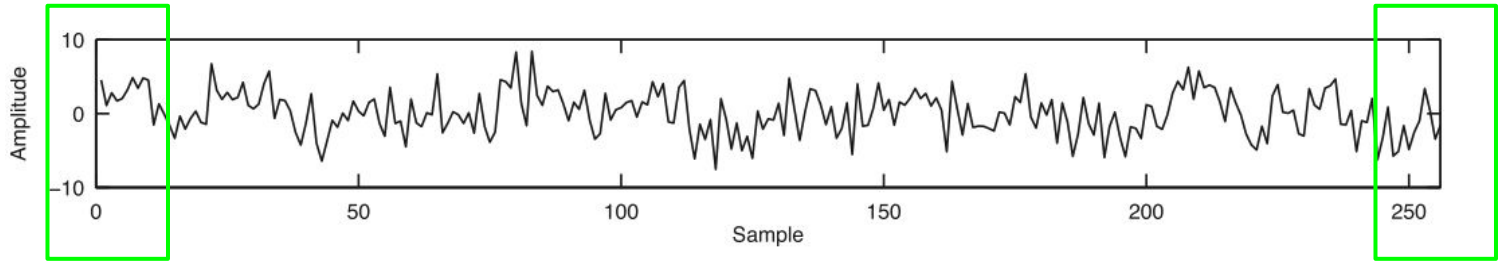
Spectral leakage

- Processed signal isn’t an integer number of periods
- Endpoints are discontinuous



Spectral leakage

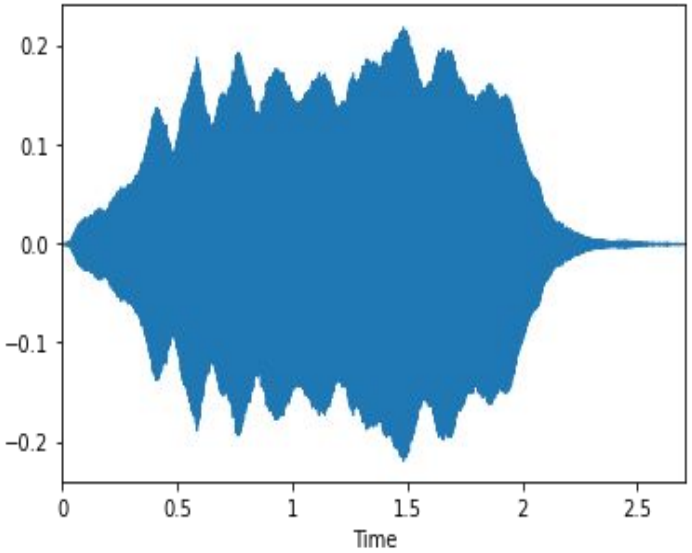
- Processed signal isn’t an integer number of periods
- Endpoints are discontinuous



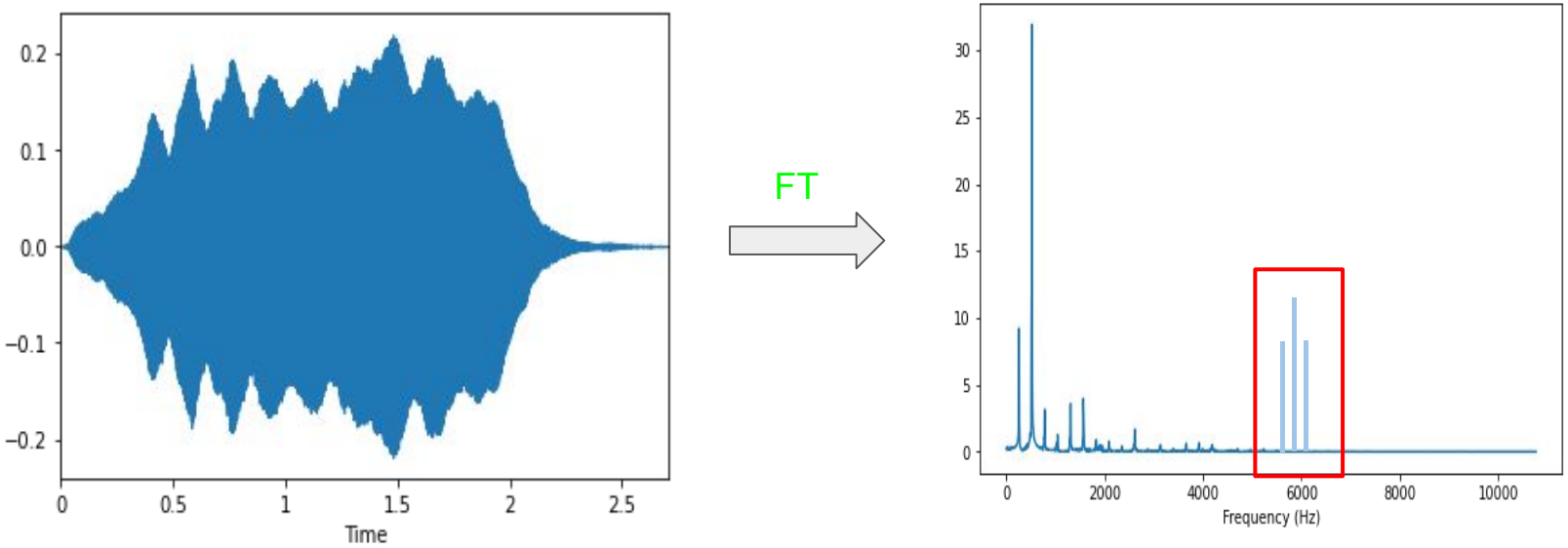
Spectral leakage

- Processed signal isn’t an integer number of periods
- Endpoints are discontinuous
- Discontinuities appear as high-frequency components not present in the original signal

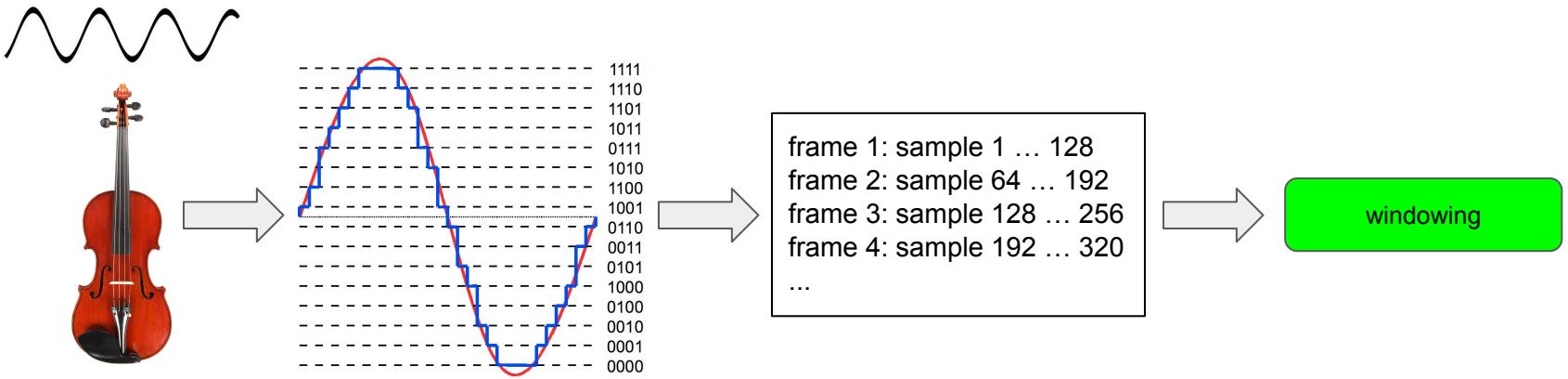
Spectral leakage



Spectral leakage



Frequency-domain feature pipeline



Windowing

- Apply windowing function to each frame

Windowing

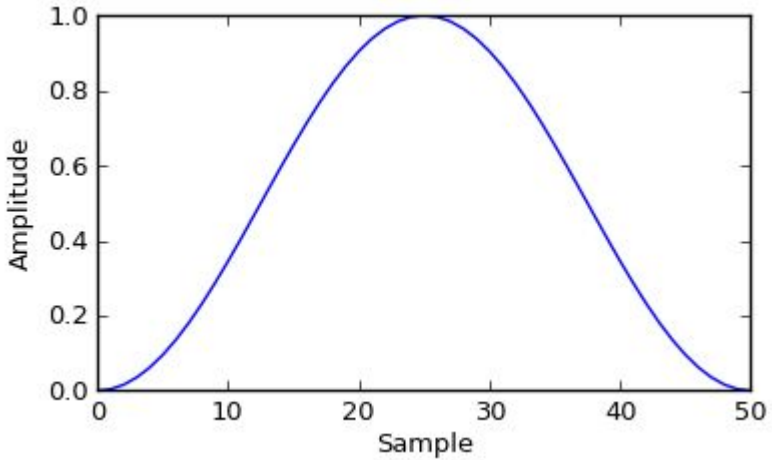
- Apply windowing function to each frame
- Eliminates samples at both ends of a frame

Windowing

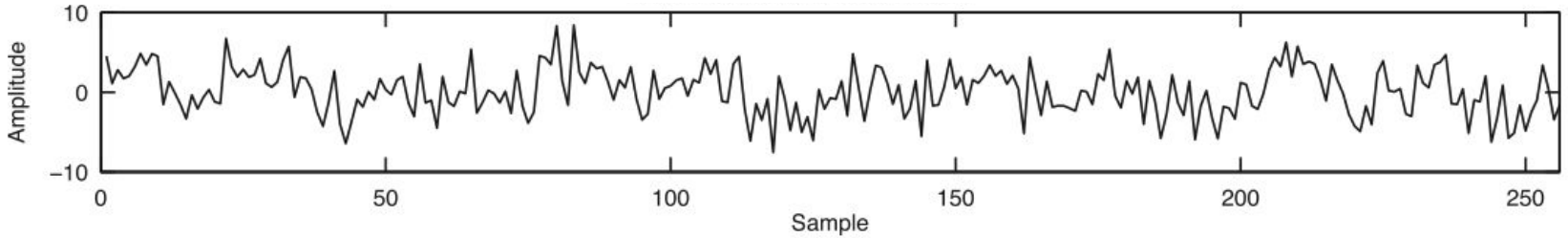
- Apply windowing function to each frame
- Eliminates samples at both ends of a frame
- Generates a periodic signal

Hann window

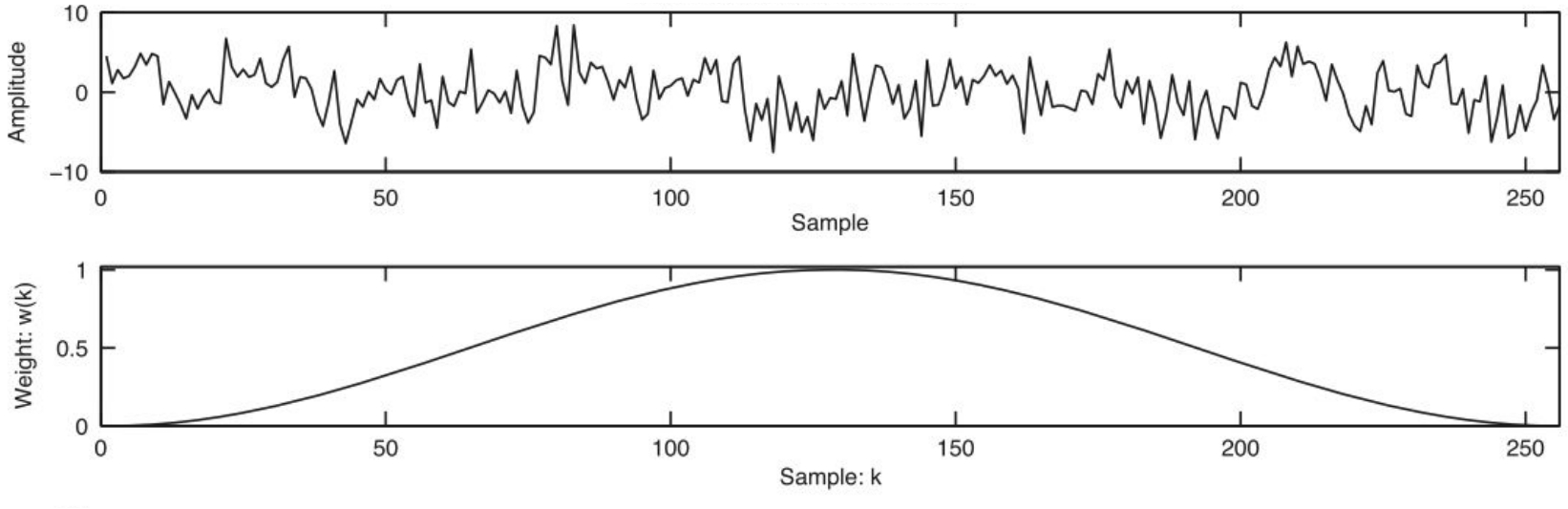
$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$



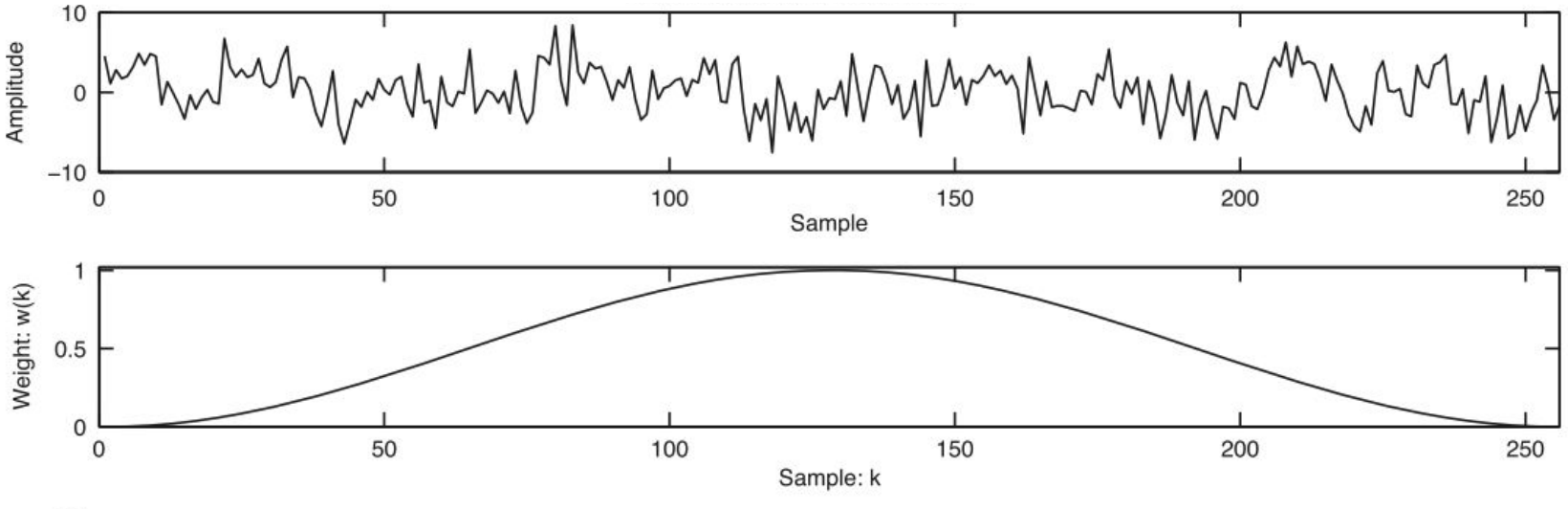
Windowing



Windowing

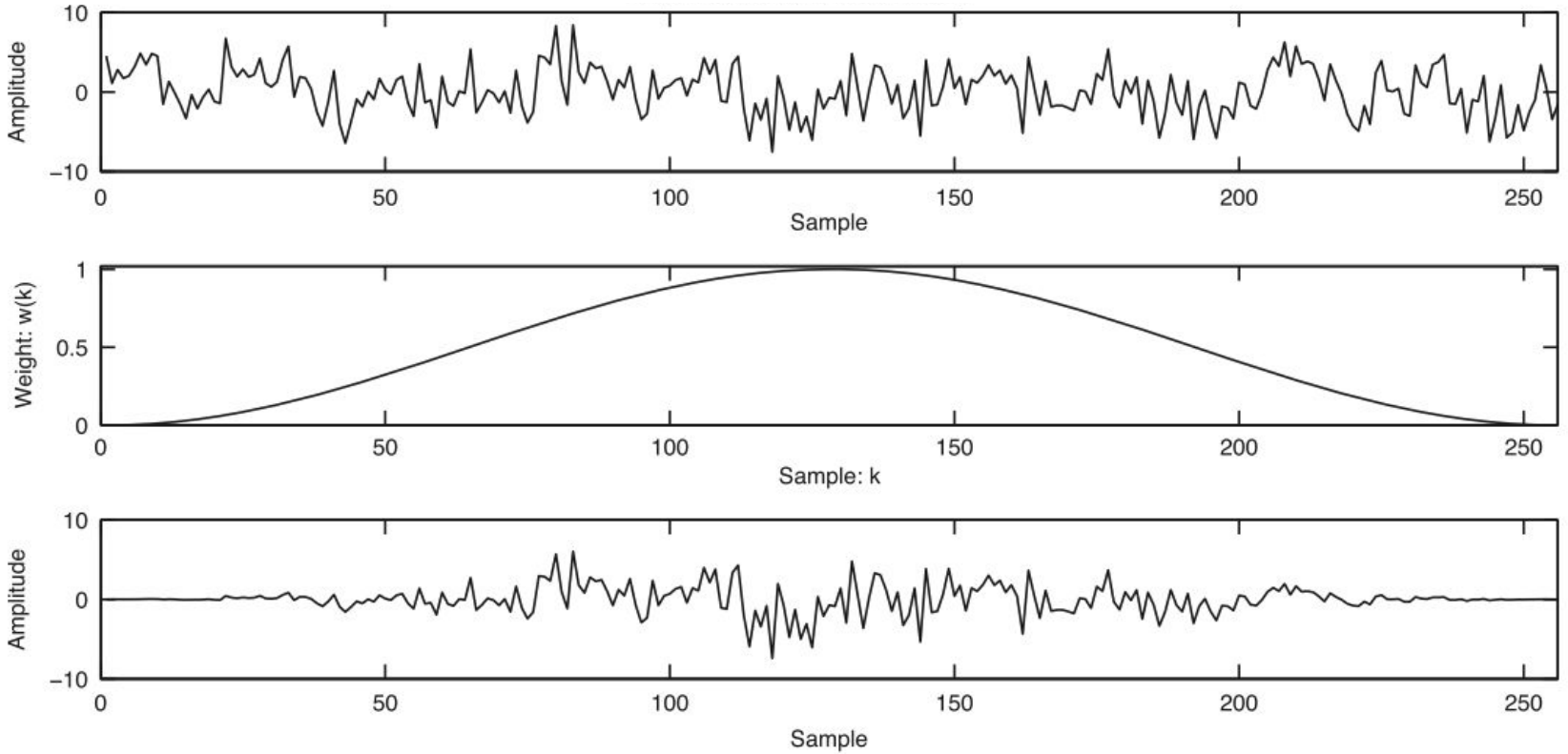

Amplitude
10
0
−10
0
50
100
150
200
250
Sample
Weight: w(k)
1
0.5
0
0
50
100
150
200
250
Sample: k

Windowing

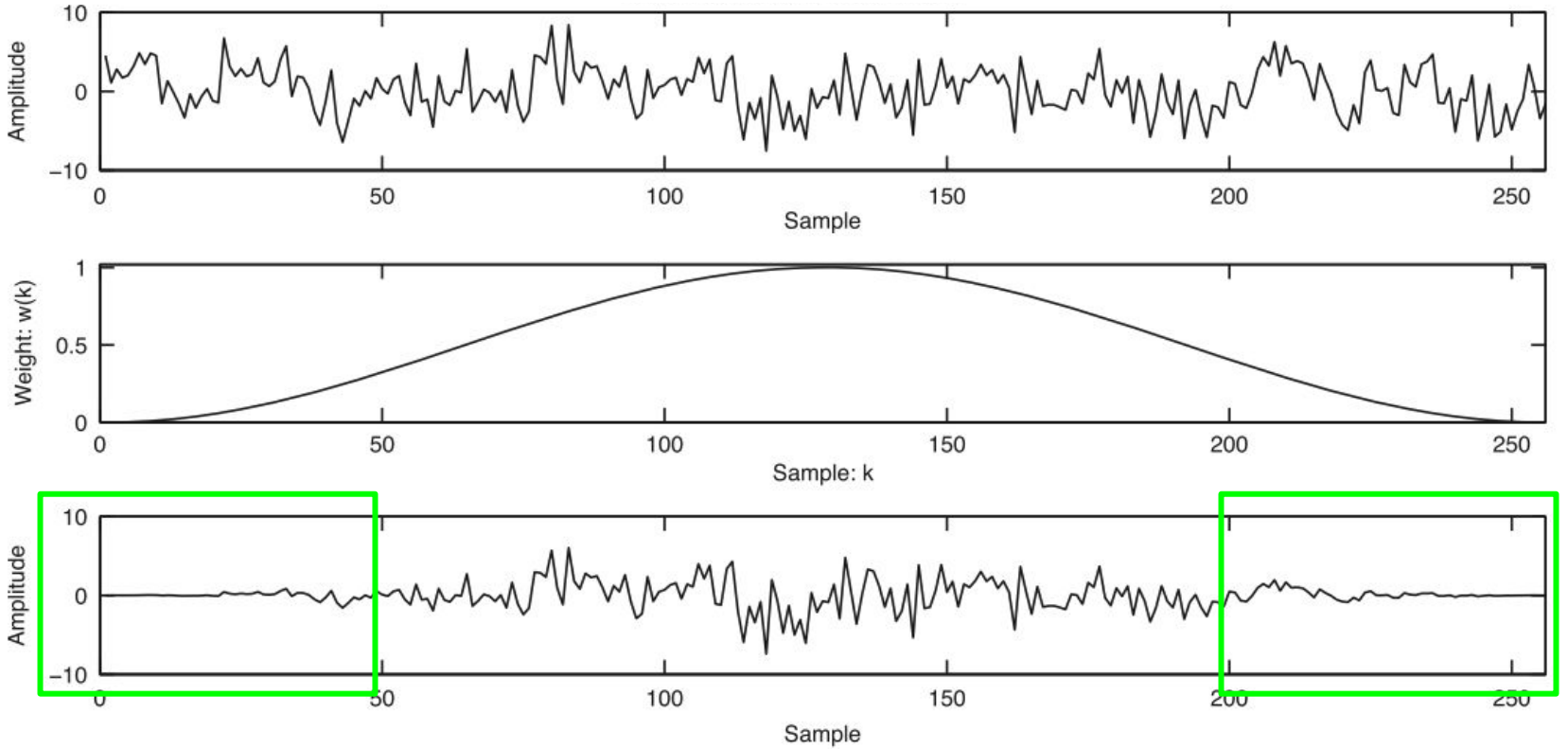


$$s_w(k) = s(k) \cdot w(k), k = 1...K$$

Windowing

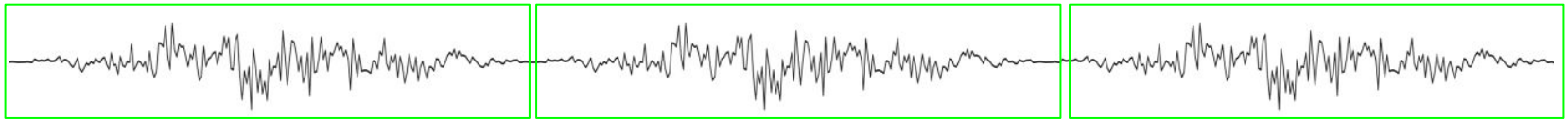

Amplitude
10
0
−10
0
50
100
150
200
250
Sample
Weight: w(k)
1
0.5
0
0
50
100
150
200
250
Sample: k
Amplitude
10
0
−10
0
50
100
150
200
250
Sample

Windowing

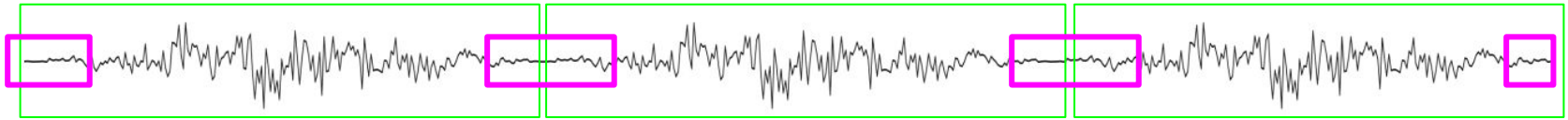

Amplitude
10
0
−10
0
50
100
150
200
250
Sample
Weight: w(k)
1
0.5
0
0
50
100
150
200
250
Sample: k
Amplitude
10
0
−10
0
50
100
150
200
250
Sample

Houston we have another problem!

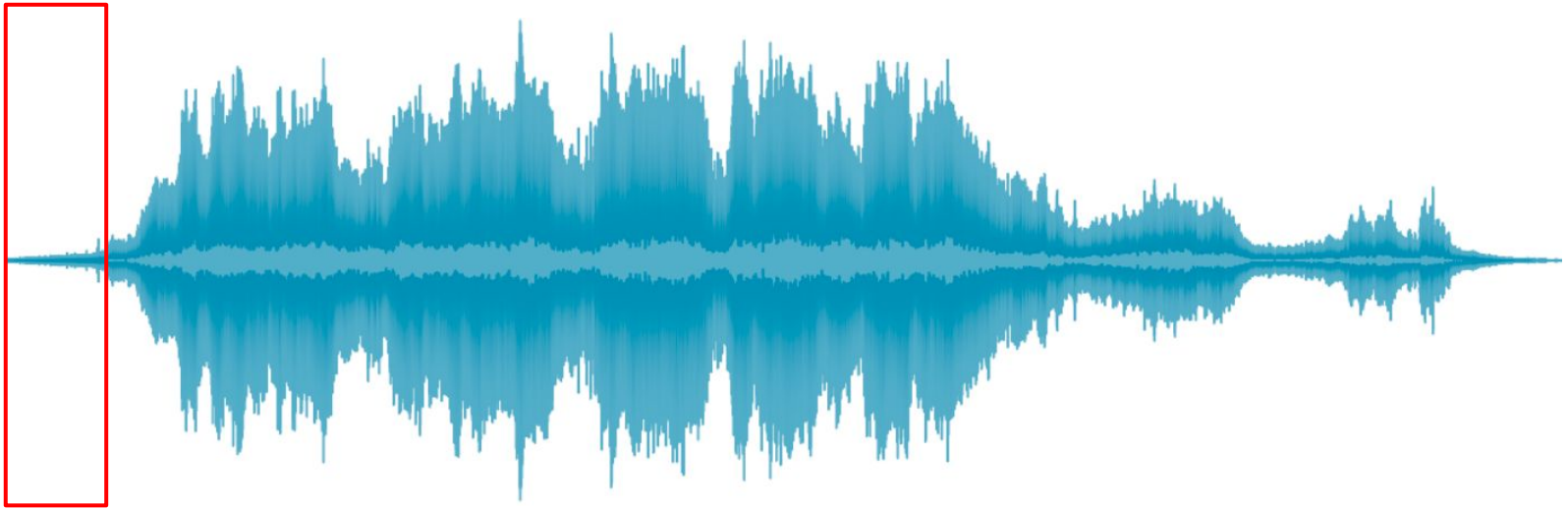
Houston we have another problem!



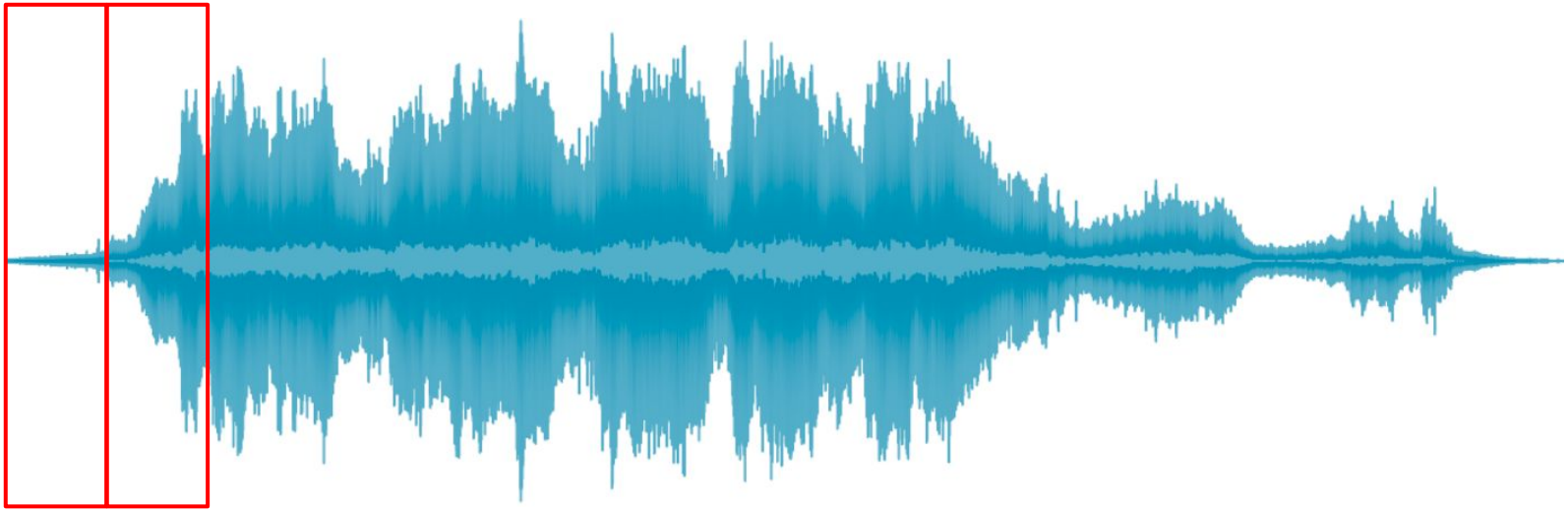
Houston we have another problem!



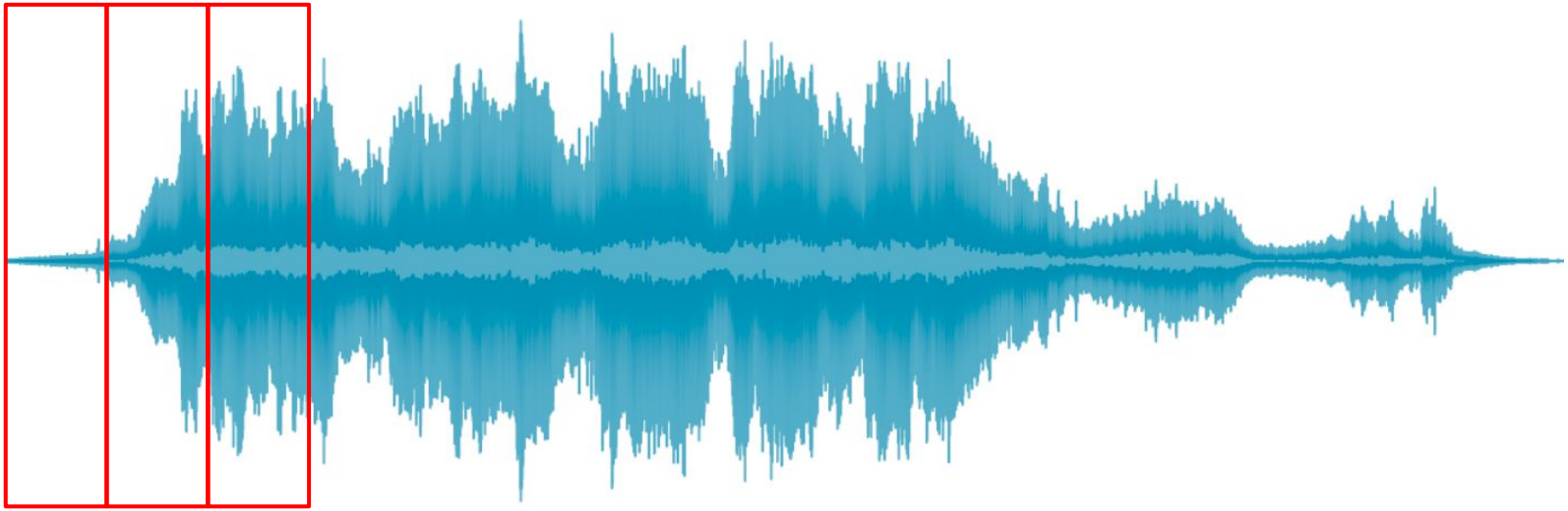
Non-overlapping frames



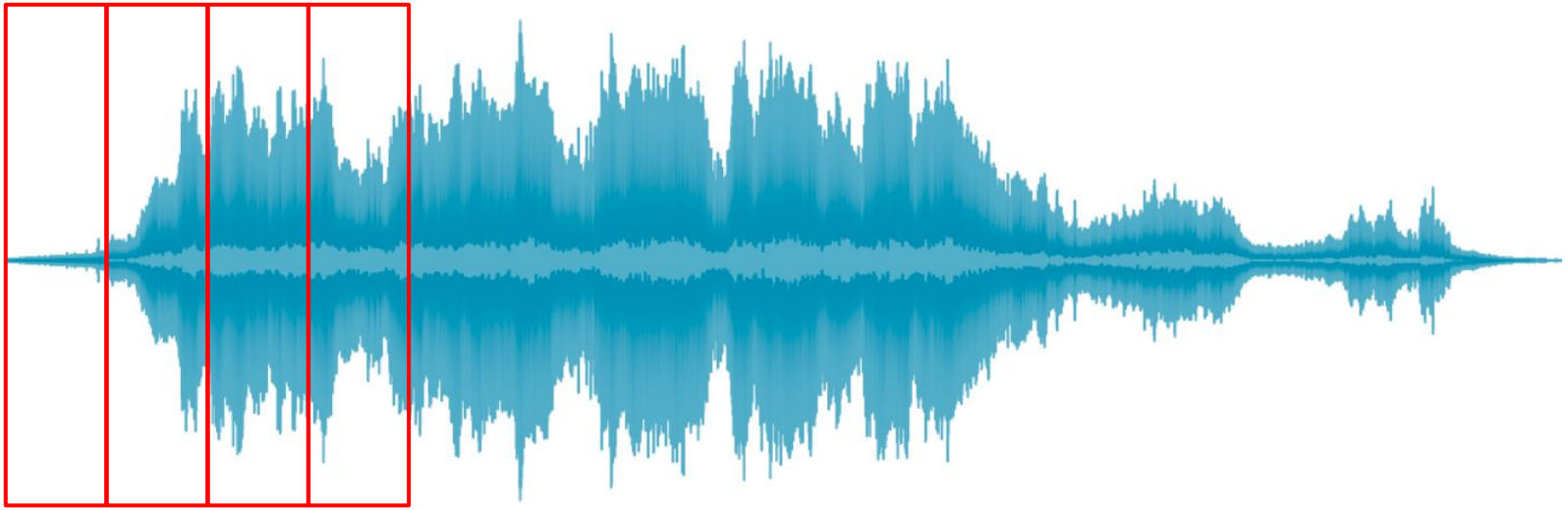
Non-overlapping frames



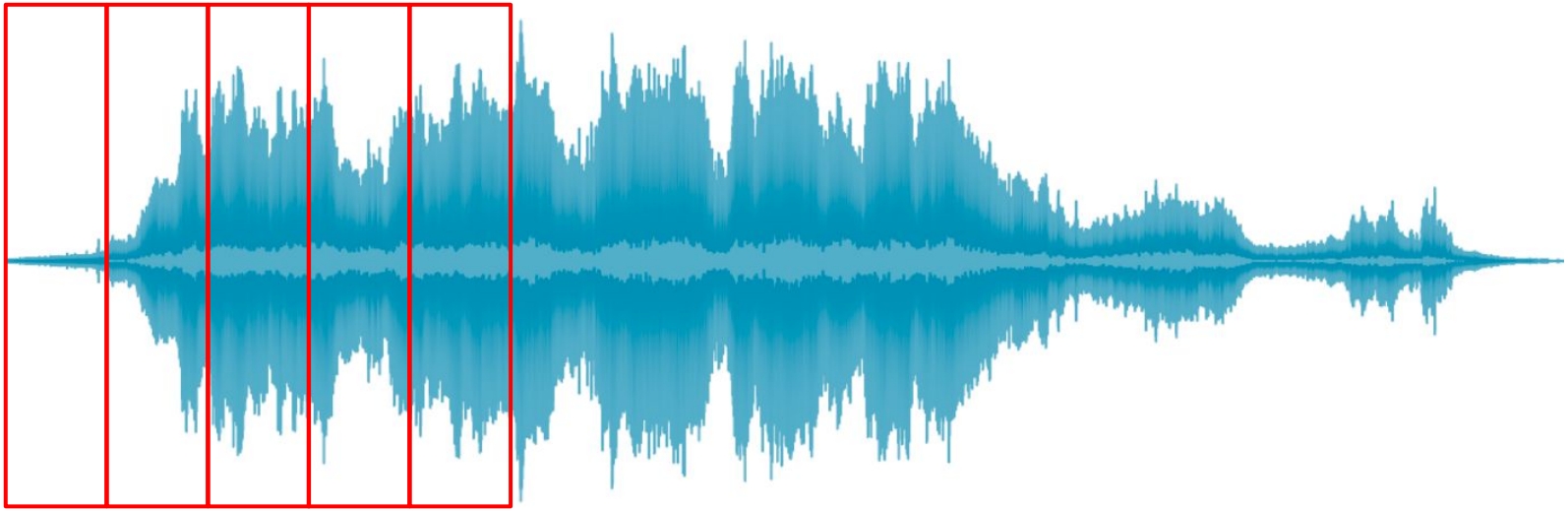
Non-overlapping frames



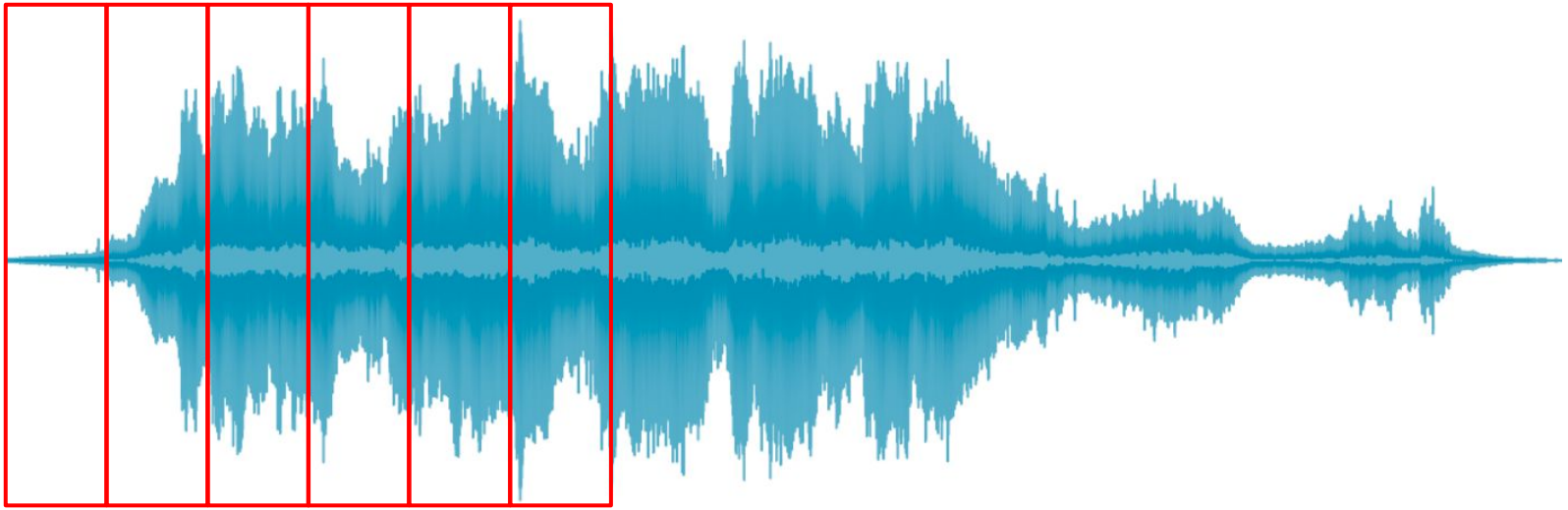
Non-overlapping frames



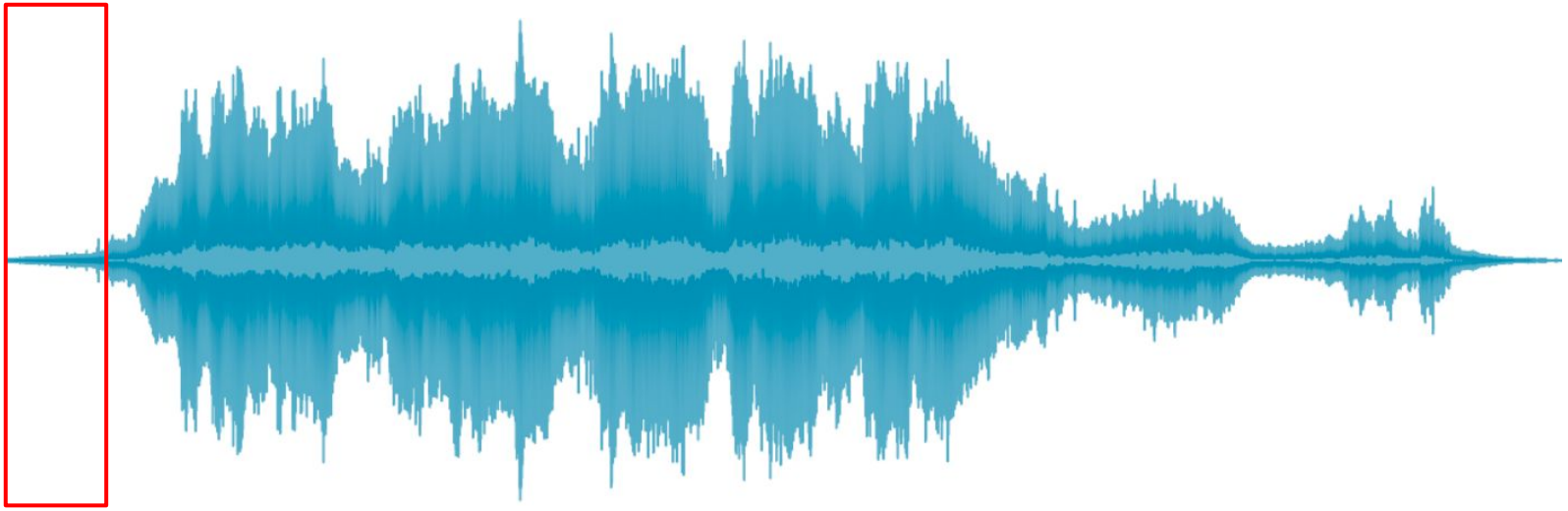
Non-overlapping frames



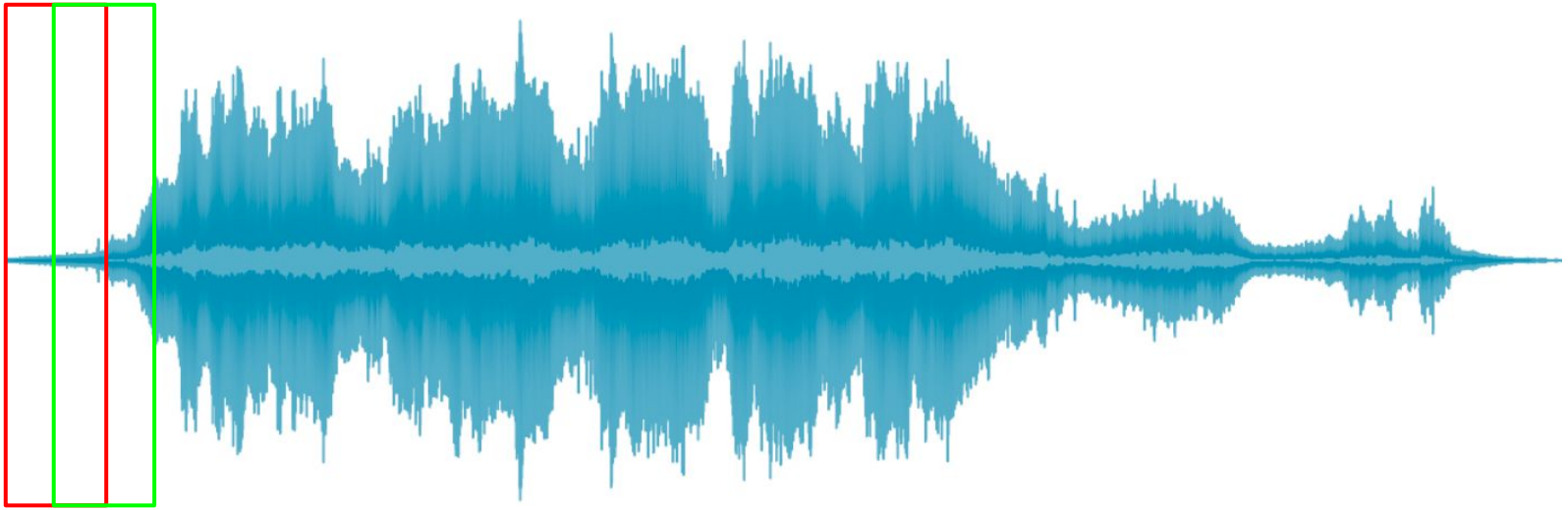
Non-overlapping frames



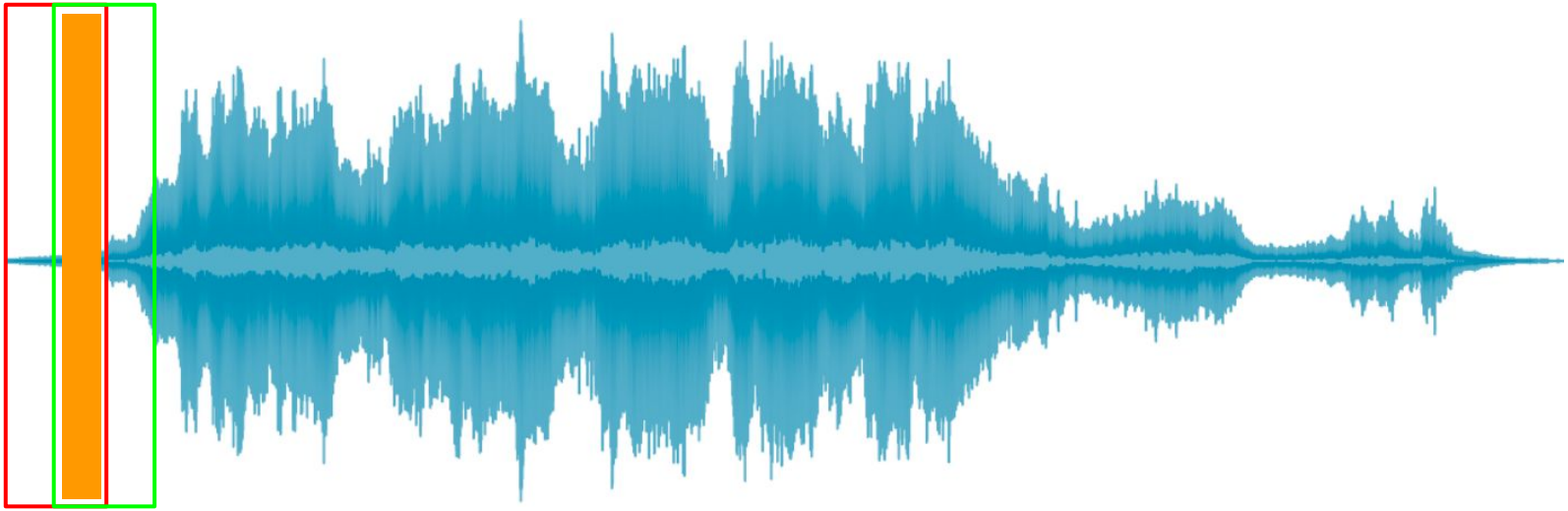
Overlapping frames



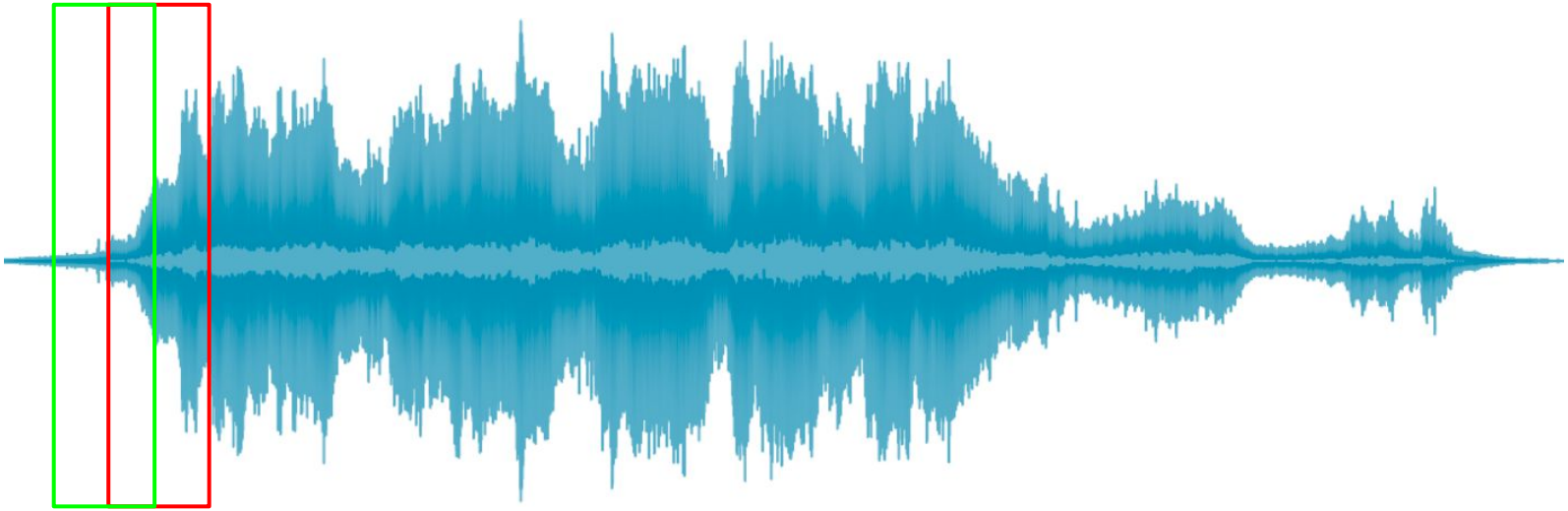
Overlapping frames



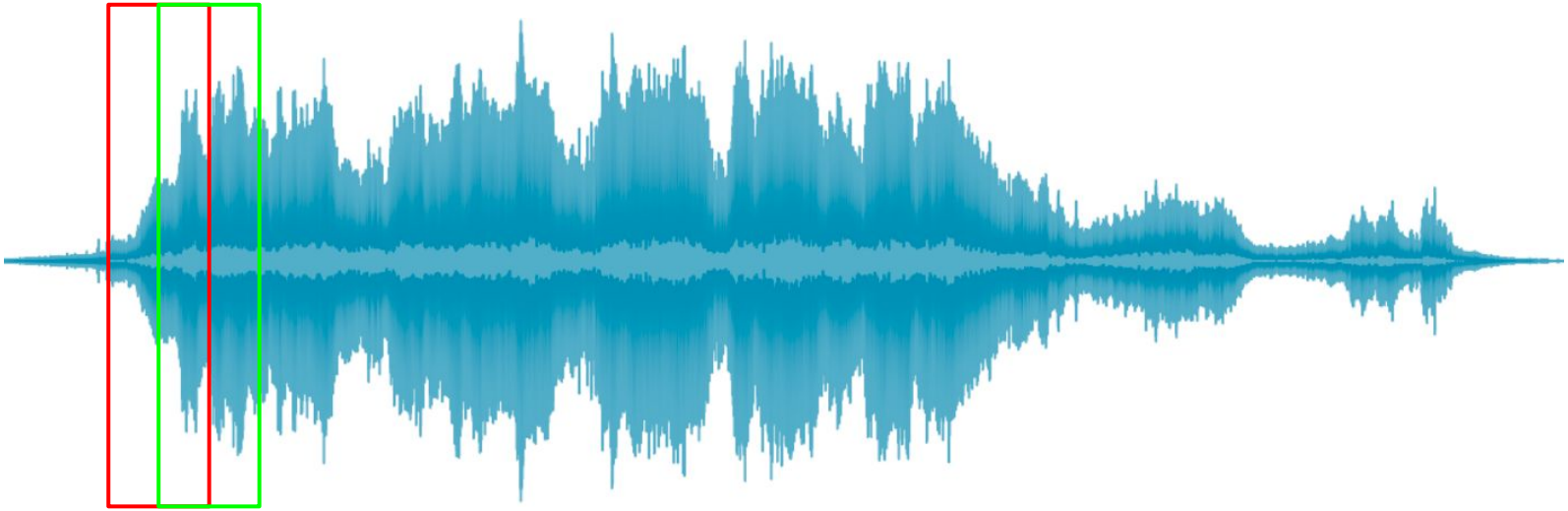
Overlapping frames



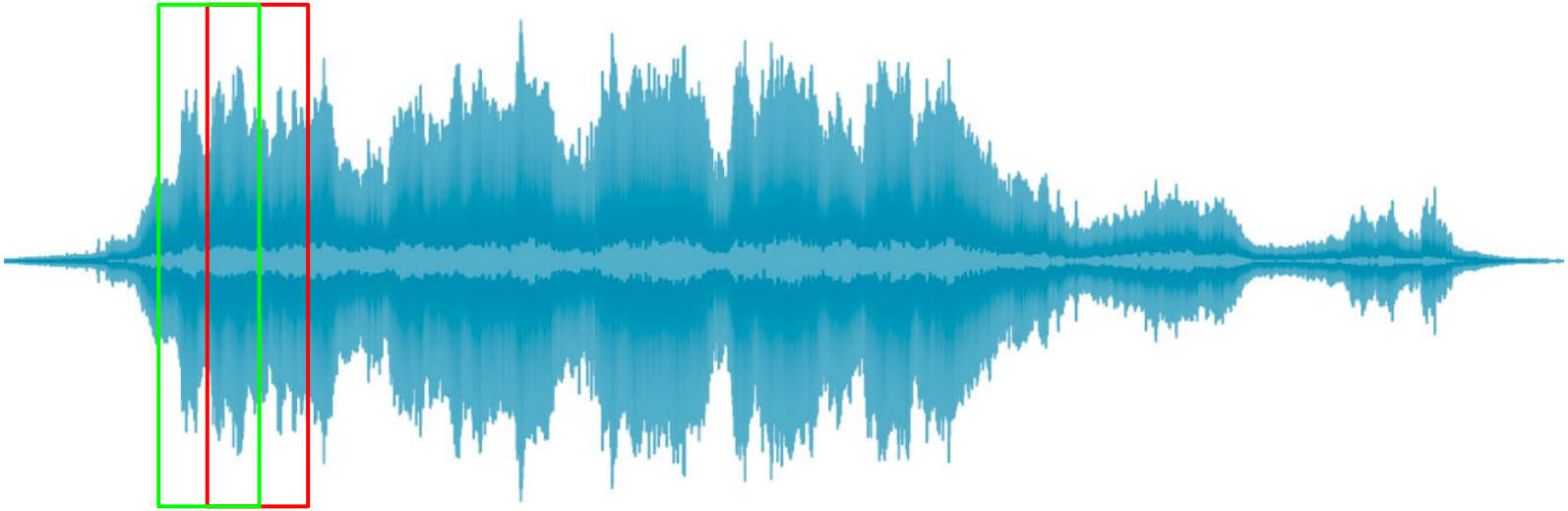
Overlapping frames



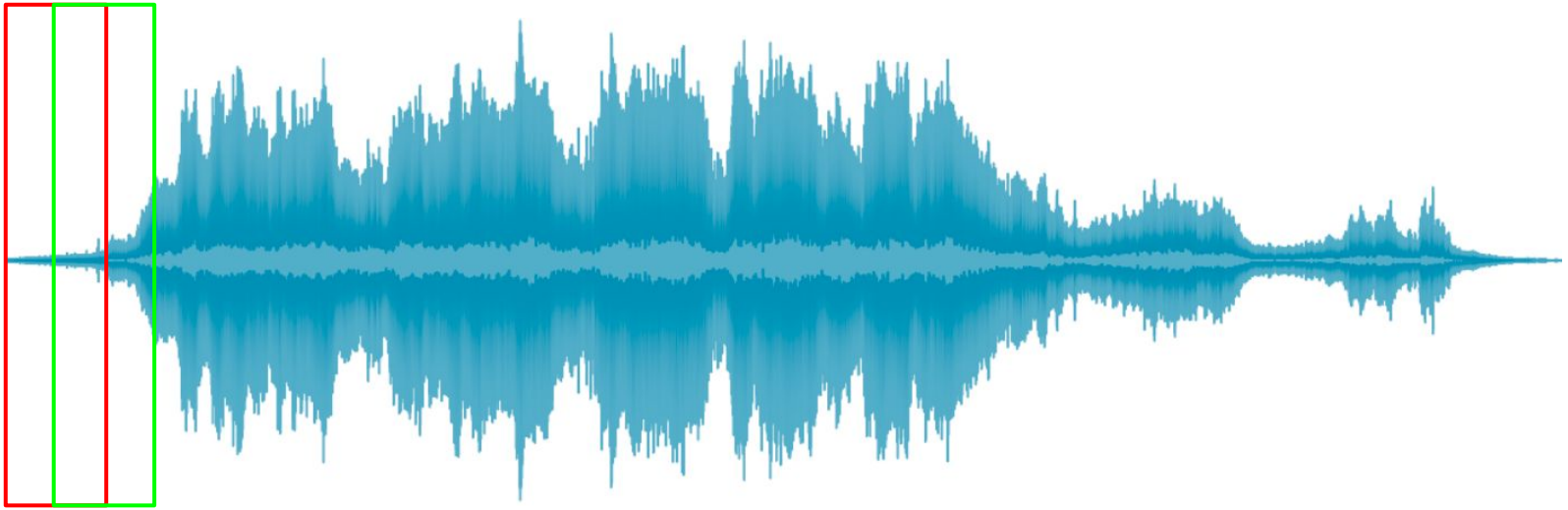
Overlapping frames



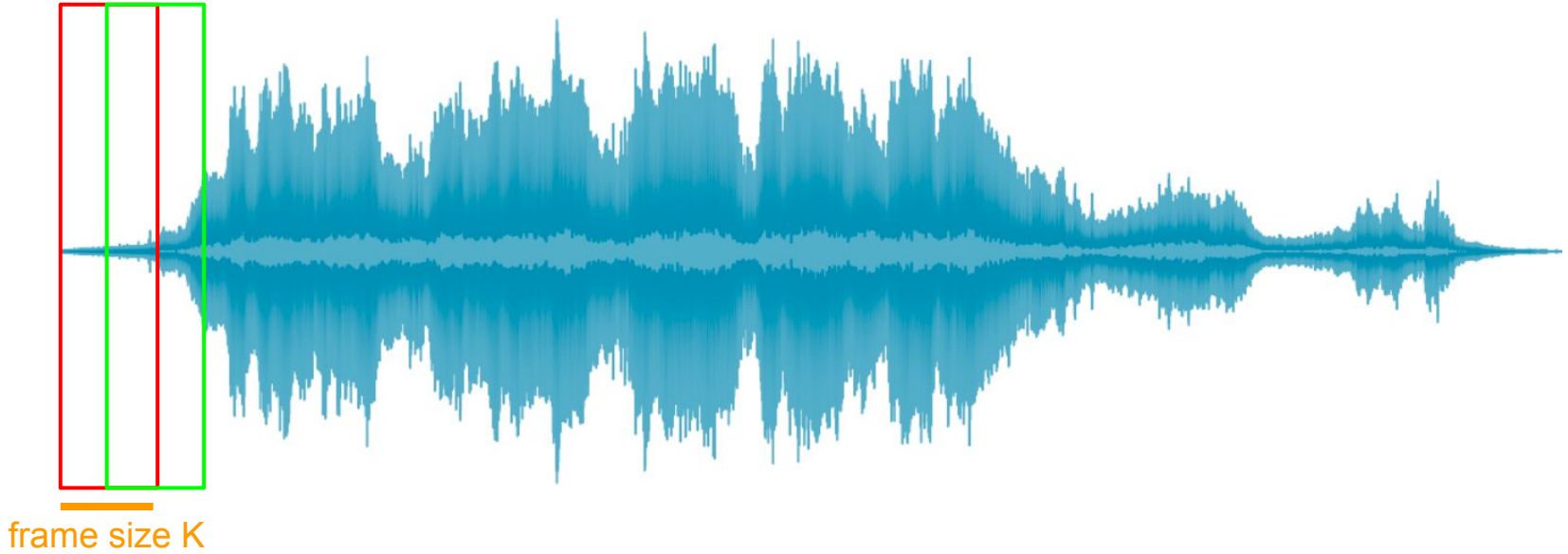
Overlapping frames



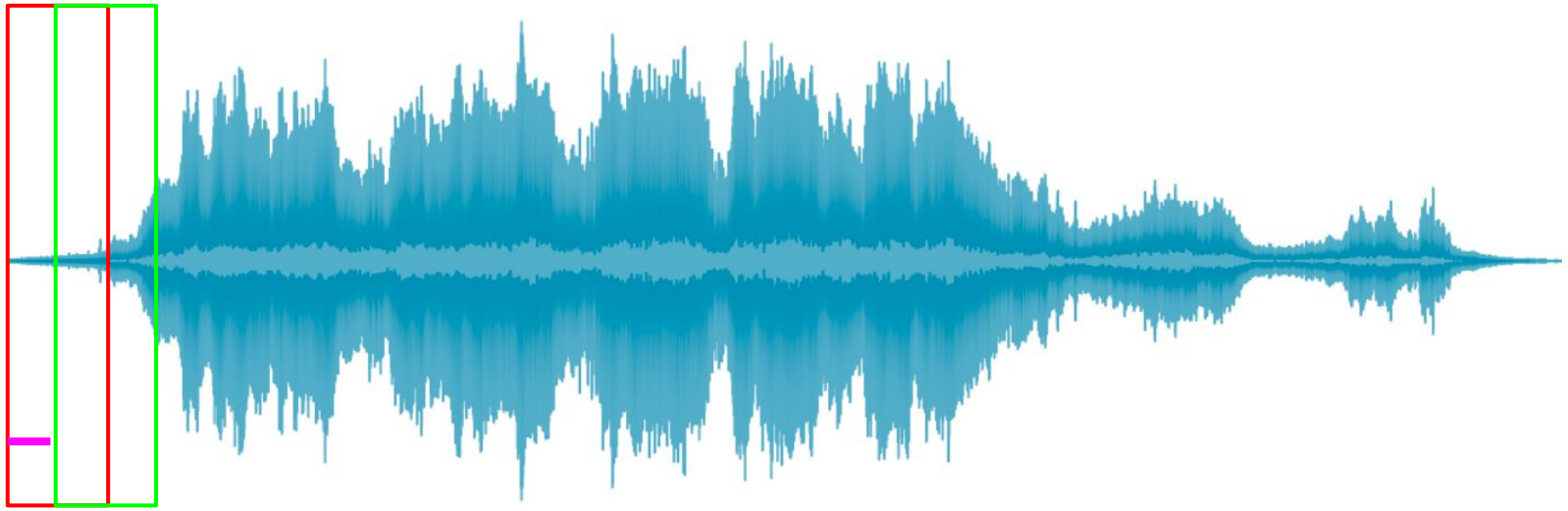
Overlapping frames



Overlapping frames

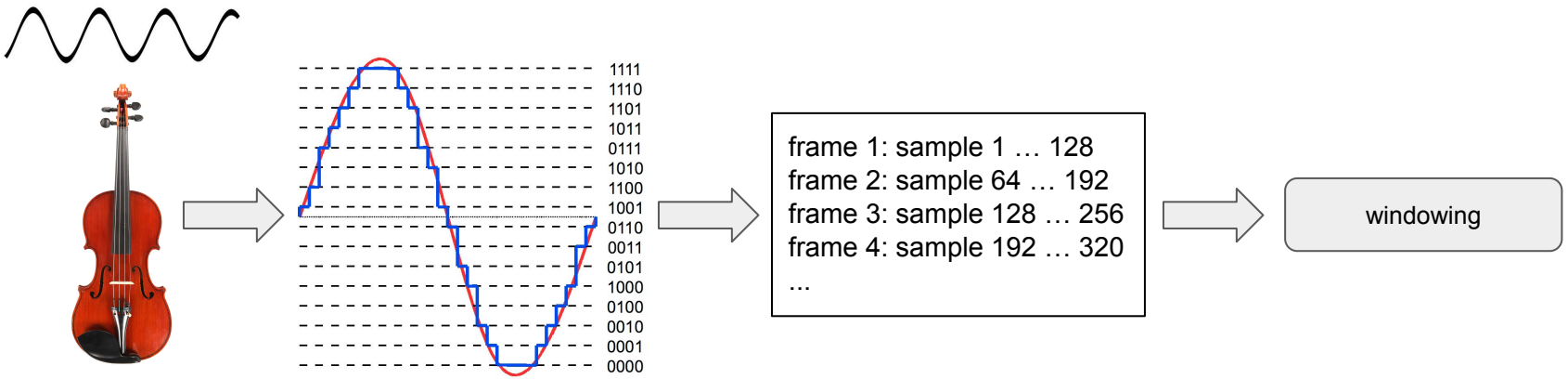


Overlapping frames

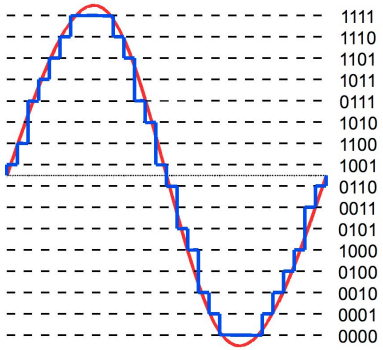


hop length

Frequency-domain feature pipeline



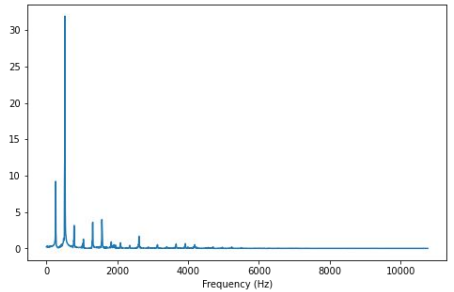
Frequency-domain feature pipeline



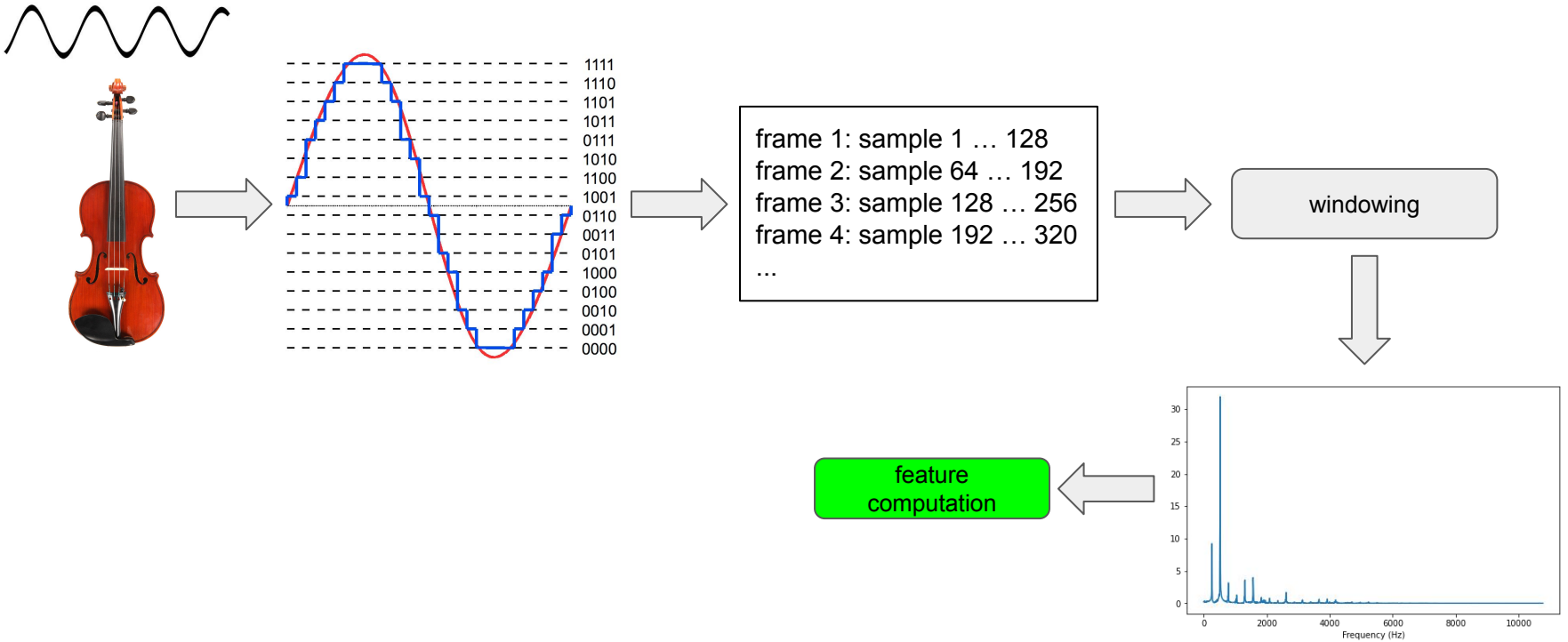
frame 1: sample 1 … 128
frame 2: sample 64 … 192
frame 3: sample 128 … 256
frame 4: sample 192 … 320
...

windowing

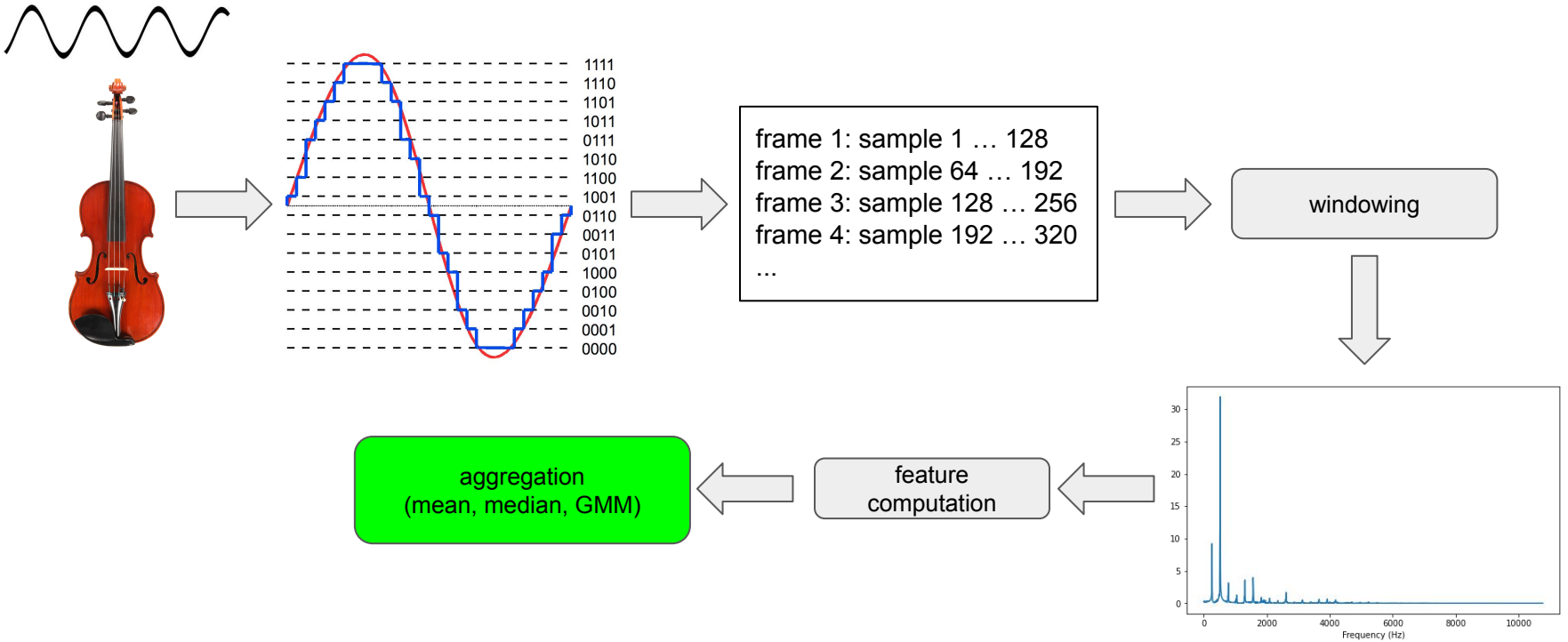
FT



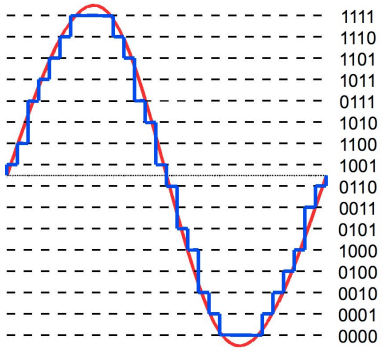
Frequency-domain feature pipeline



Frequency-domain feature pipeline

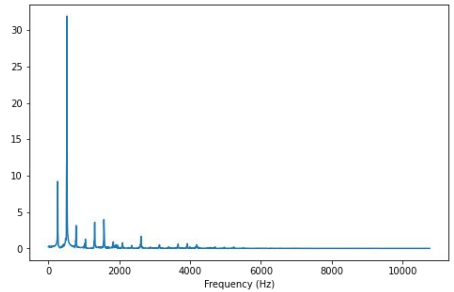


Frequency-domain feature pipeline



frame 1: sample 1 … 128
frame 2: sample 64 … 192
frame 3: sample 128 … 256
frame 4: sample 192 … 320
...

windowing



feature computation

aggregation (mean, median, GMM)

feature value/vector/matrix

What’s up next?

- Time-domain features