PAT 498/598 (Winter 2025)

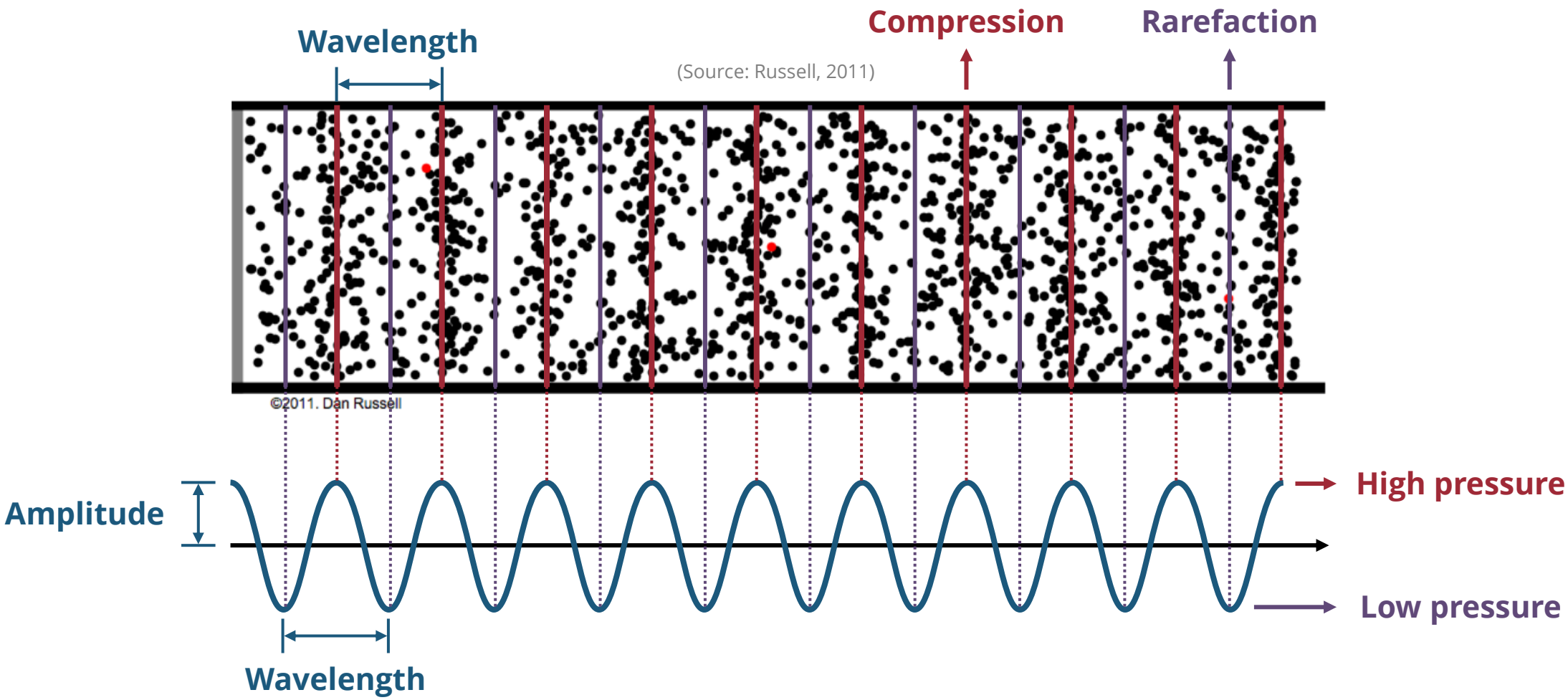
Music & AI

Lecture 6: Spectral Analysis

Instructor: Hao-Wen Dong

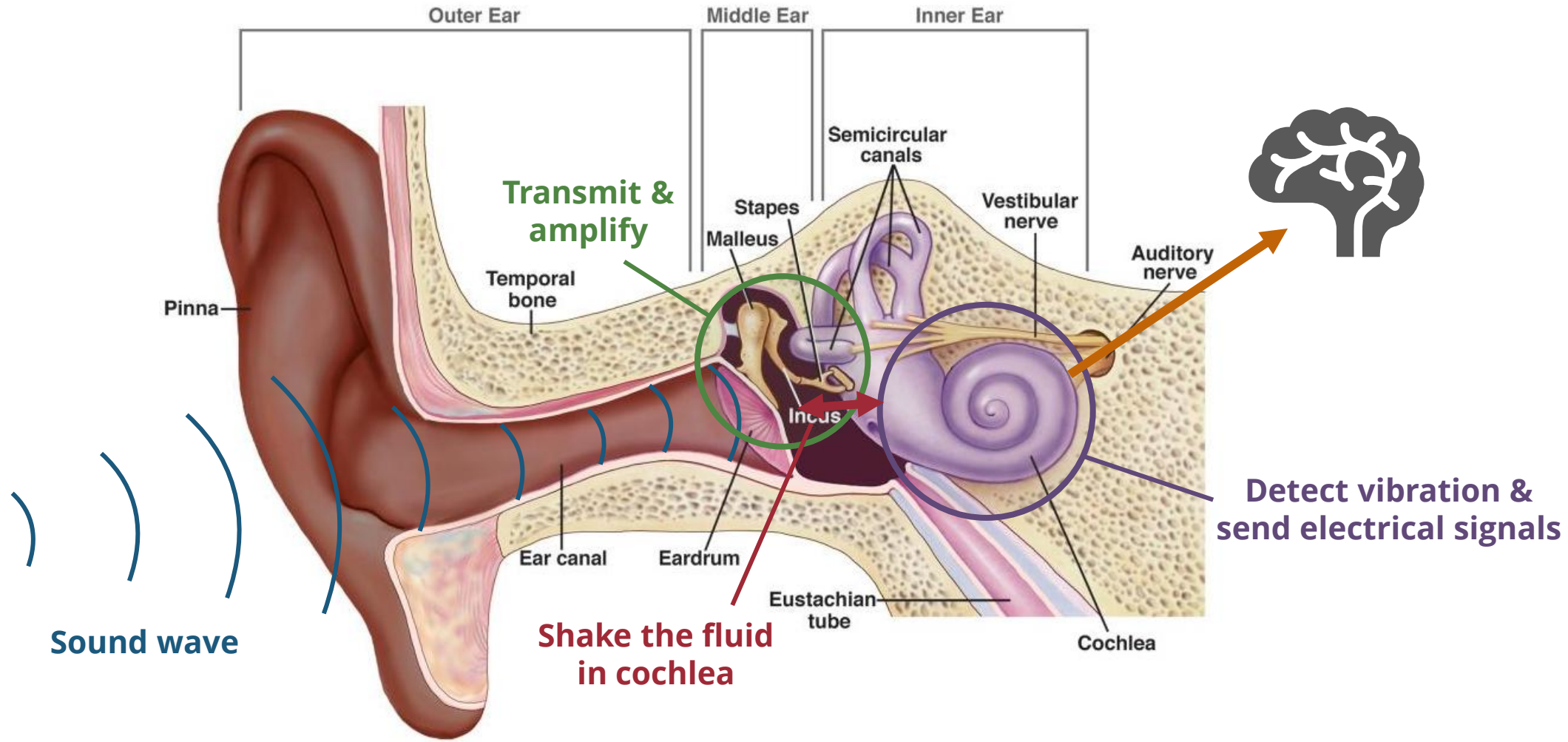
SCHOOL OF MUSIC, THEATRE & DANCE
PERFORMING ARTS TECHNOLOGY
UNIVERSITY OF MICHIGAN

(Recap) Longitudinal vs Transverse Waves



Dan Russell, "Longitudinal and Transverse Wave Motion," *Acoustics and Vibration Animations*, 1998 (updated on 2016).

(Recap) Human Ears



(Source: NIH/NIDCD)

National Institute on Deafness and Other Communication Disorders (NIDCD), "How Do We Hear?," National Institutes of Health (NIH), 2015.

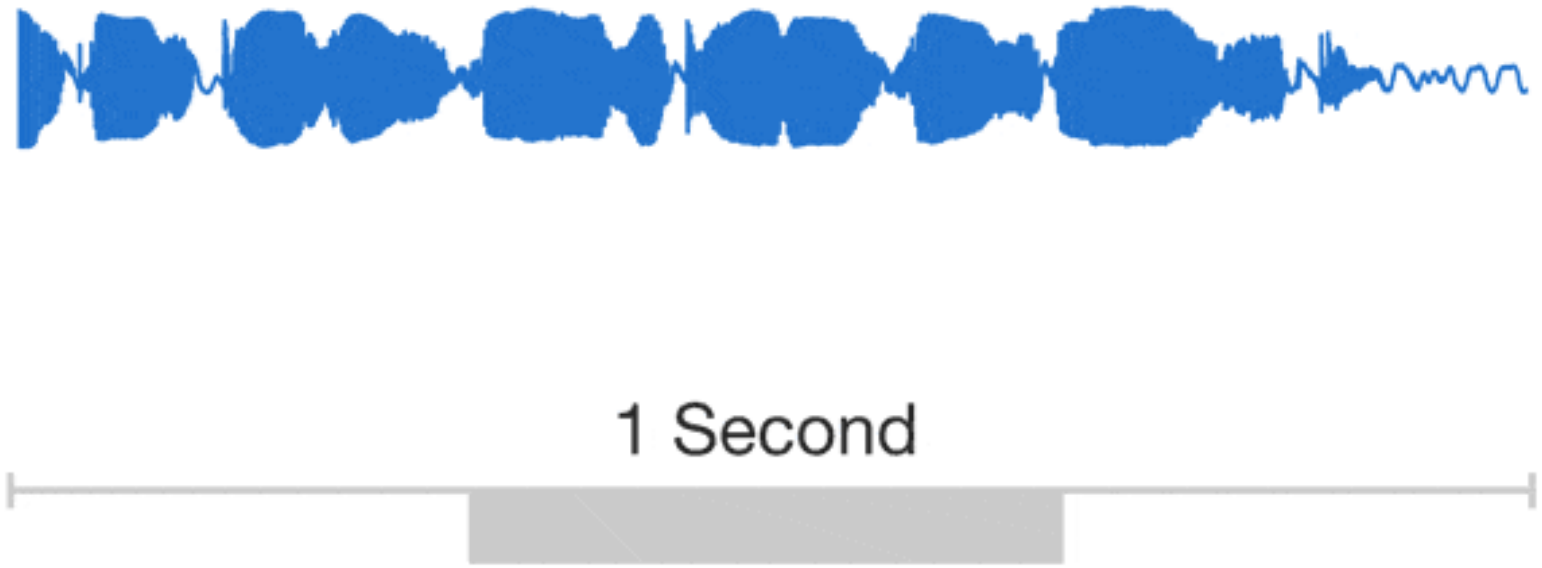
(Recap) Sound Intensity & Decibels

- **Sound intensity** is defined as the sound power per unit area
  - Usually measured in **watt per square meter** (W/m^2)
- **Sound intensity *level*** is defined as

$$I_{\text{dB}} \coloneqq 10 \log_{10}\left(\frac{I}{I_{REF}}\right)$$

  - $I_{REF} \coloneqq 10^{-12} W/m^2$ is the **threshold of hearing** (TOH)
  - TOH: minimum sound intensity of a pure tone that a human can hear

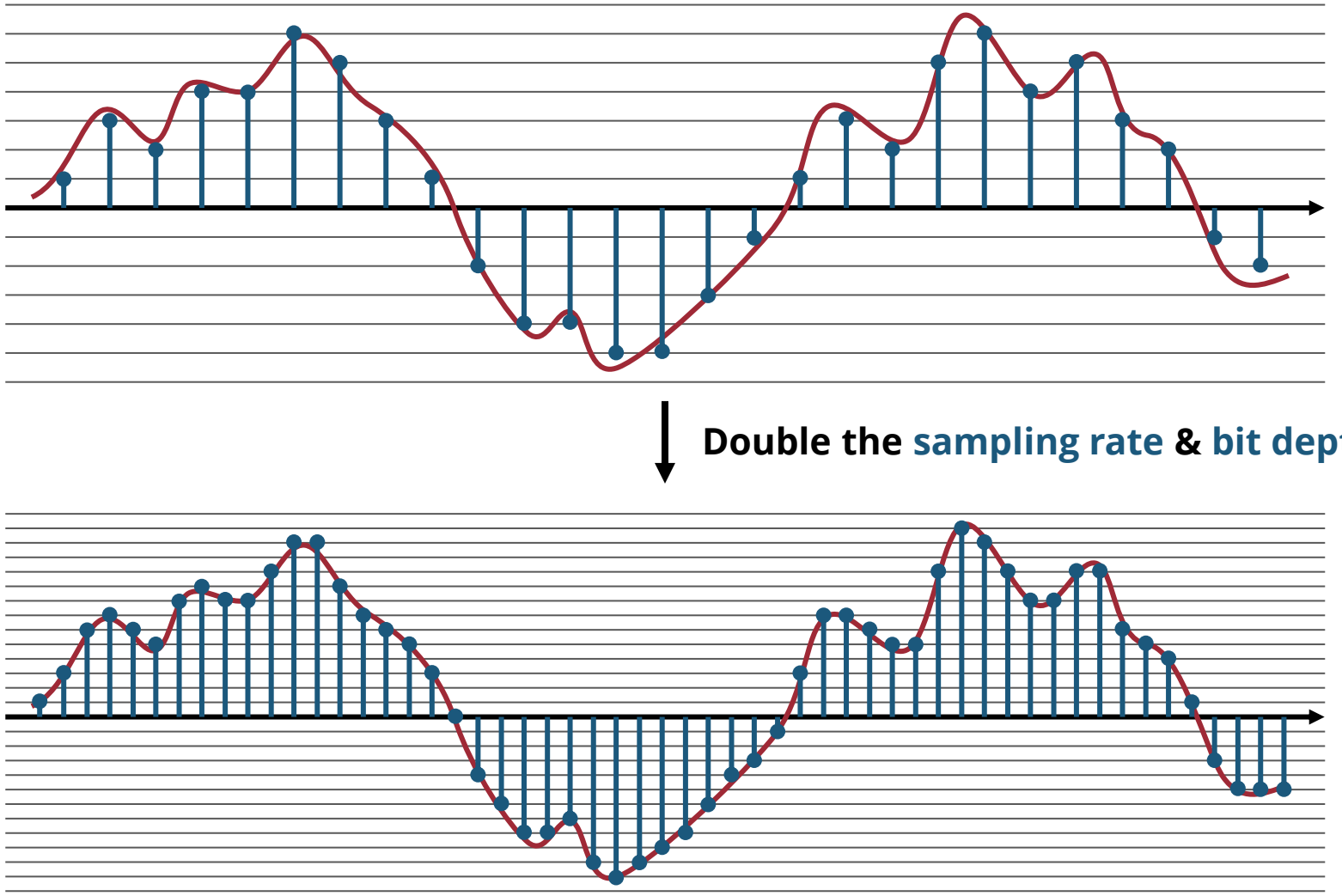
(Recap) Digital Audio



(Source: van den Oord et al., 2016)

Aaron van den Oord and Sander Dieleman, "WaveNet: A generative model for raw audio," *DeepMind Blog*, September 8, 2016.

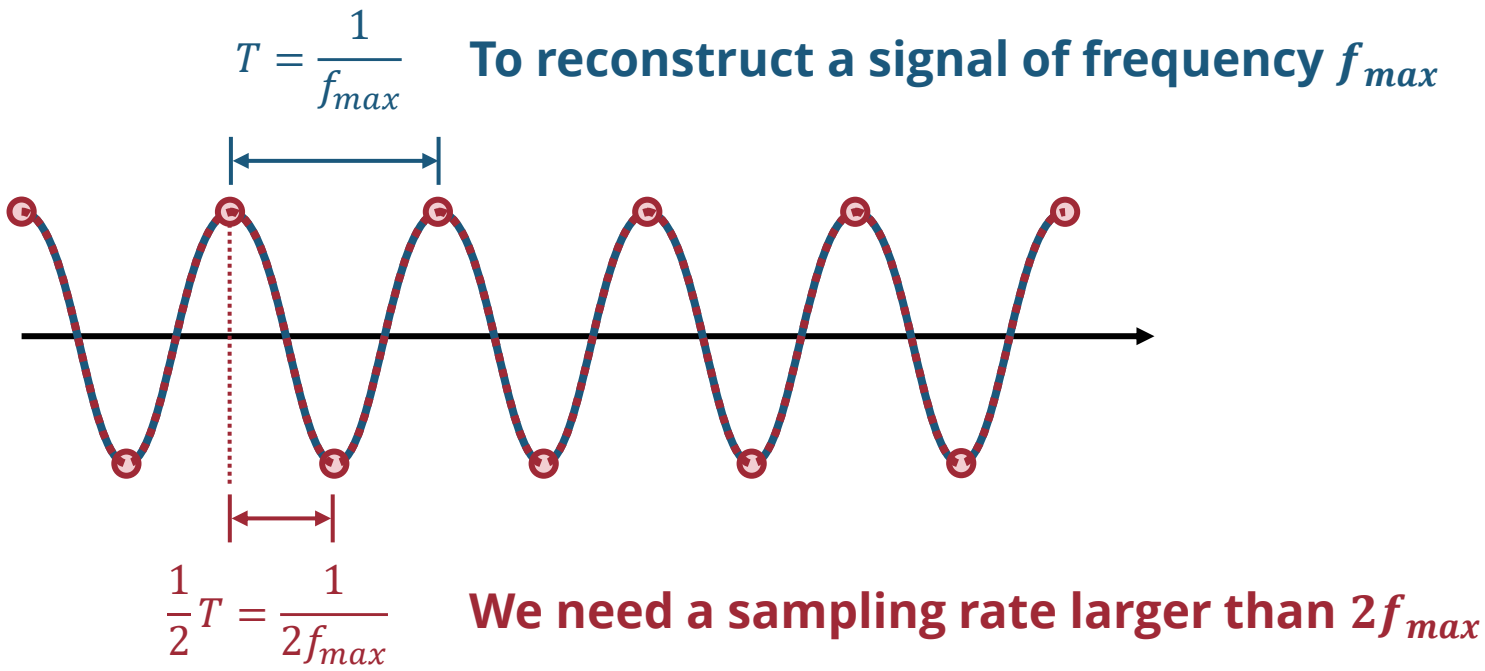
(Recap) Resolution: Sampling Rate & Bit Depth



Sampling Theorem

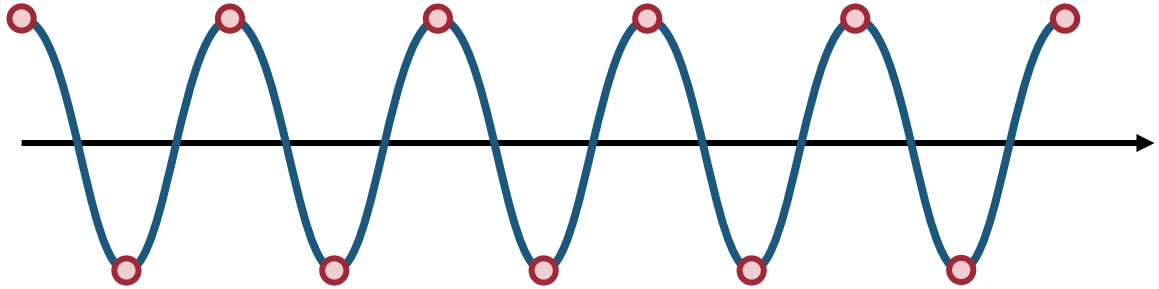
Nyquist–Shannon Sampling Theorem

- **Theorem**: If a signal contains no frequencies higher than f_{max}, then the signal can be perfectly reconstructed when sampled at a rate $f_s > 2f_{max}$
  - $2f_{max}$ is usually referred to as the **Nyquist rate**

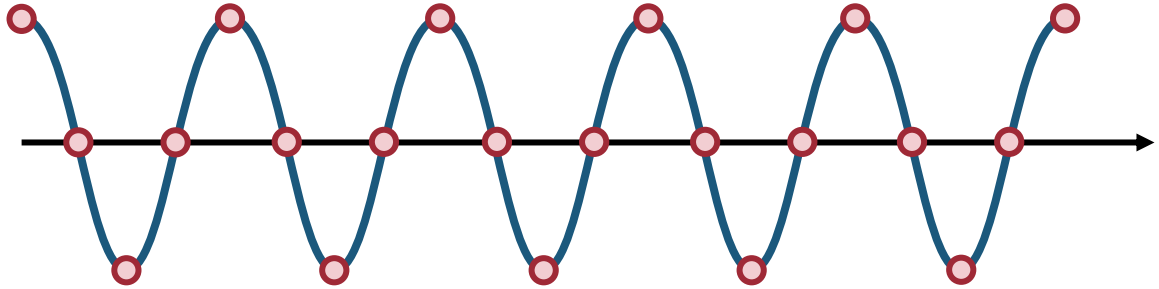


Sampling Theorem: Oversampling

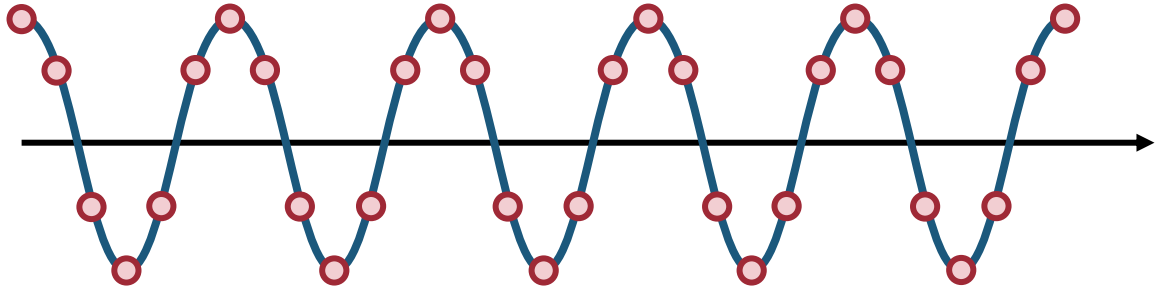
Critically sampled
$(f_s = 2f_{max})$



Oversampled
$(f_s = 4f_{max})$



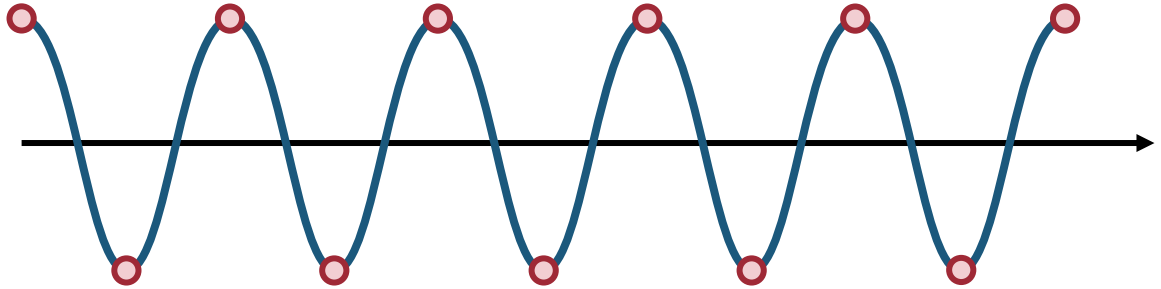
Oversampled
$(f_s = 6f_{max})$



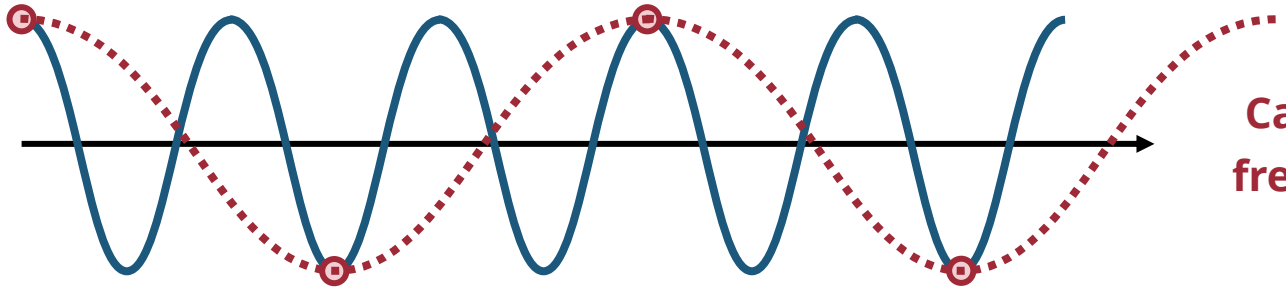
Reconstruction is possible!

Sampling Theorem: Undersampling

Critically sampled
$(f_s = 2f_{max})$

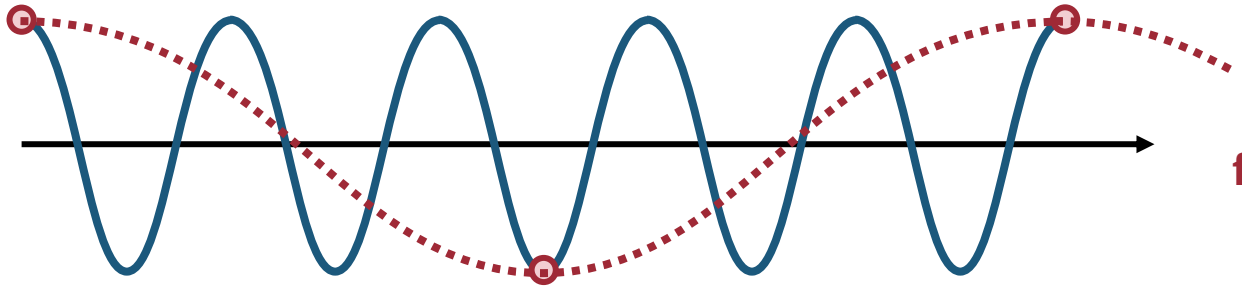


Undersampled
$(f_s = \frac{2}{3} f_{max})$



Can only reconstruct frequency up to $\frac{1}{3} f_{max}$

Undersampled
$(f_s = \frac{2}{5} f_{max})$



Can only reconstruct frequency up to $\frac{1}{3} f_{max}$

Sampling Theorem

- Telephone audio is sampled at 8 kHz. What is the maximum frequency it can reconstruct?
- To cover the **human hearing range (20Hz to 20 kHz)**, what is the minimum sampling rate required?

Sampling Rate & Frequency Range

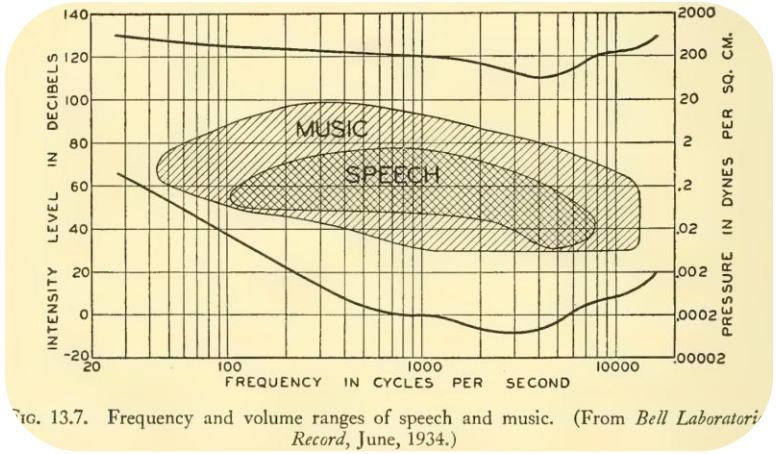
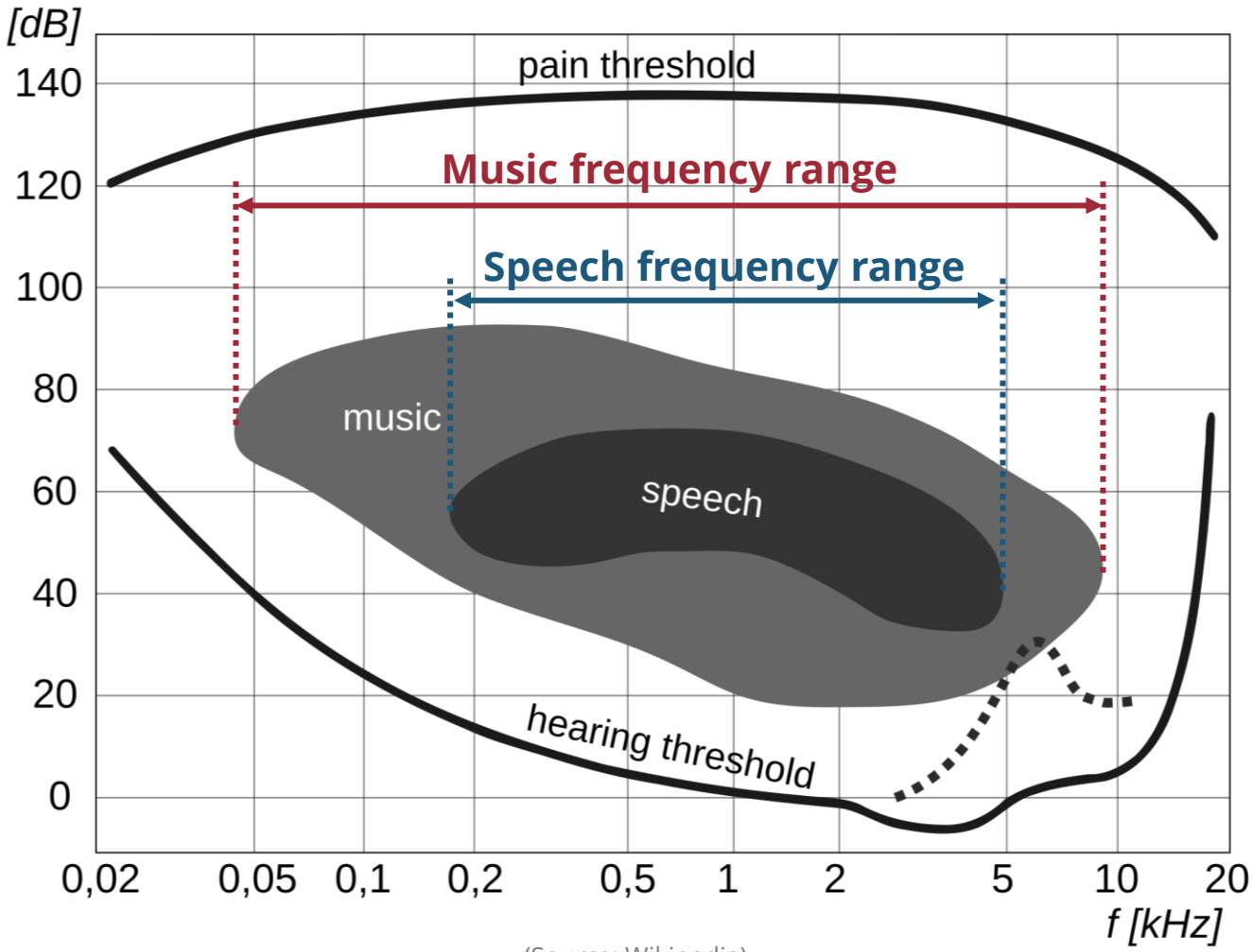


FIG. 13.7. Frequency and volume ranges of speech and music. (From *Bell Laboratori Record*, June, 1934.)

(Source: Bell Laboratories Record 1934 & Olson 1947)



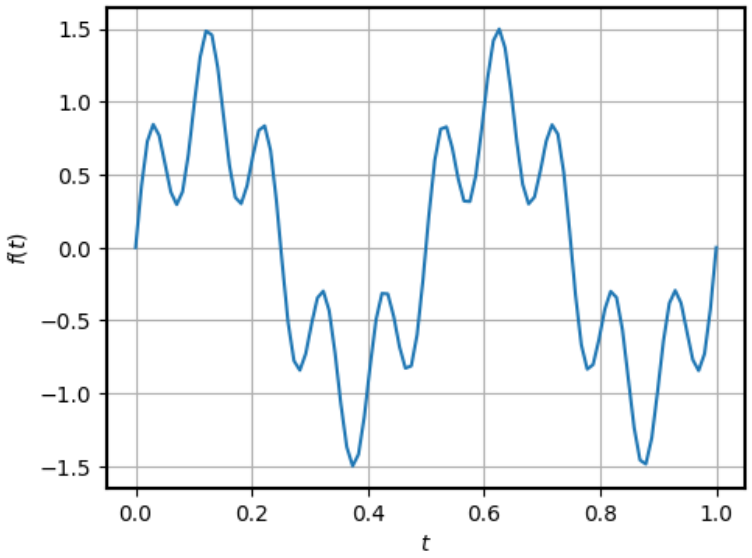
(Source: Wikipedia)

Bell Laboratories Record, 12(6):314, 1934.
Harry Ferdinand Olson, "Speech, Music and Hearing," *Elements of acoustical engineering Hardcover*, p. 326, 1947.
en.wikipedia.org/wiki/Hearing_range

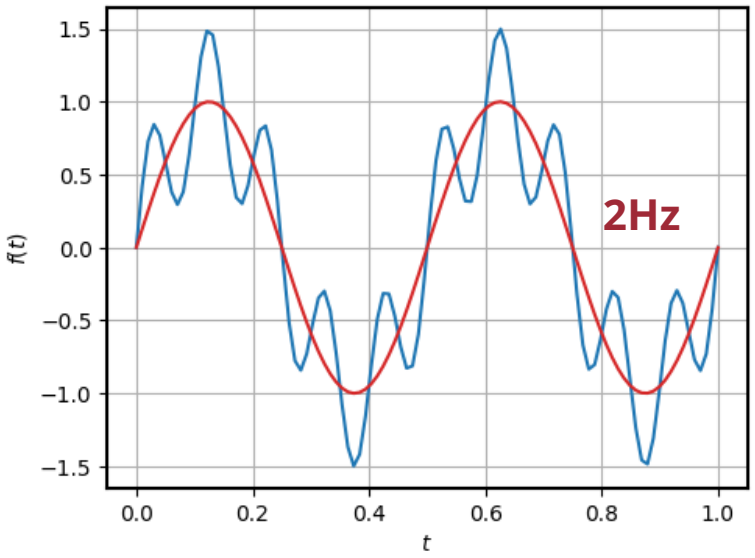
Spectral Analysis

Spectral Analysis

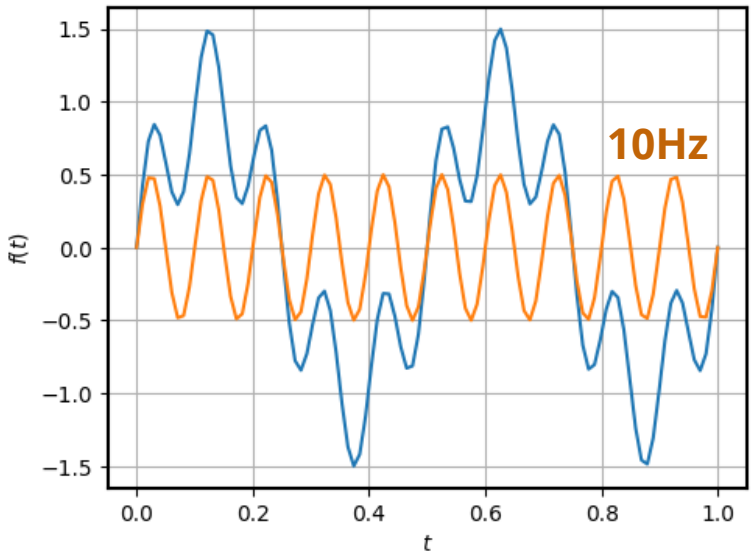
- **Goal**: Analyze the **frequency components** of a signal



$$\sin(2 \cdot 2\pi t) + \frac{1}{2}\sin(10 \cdot 2\pi t)$$

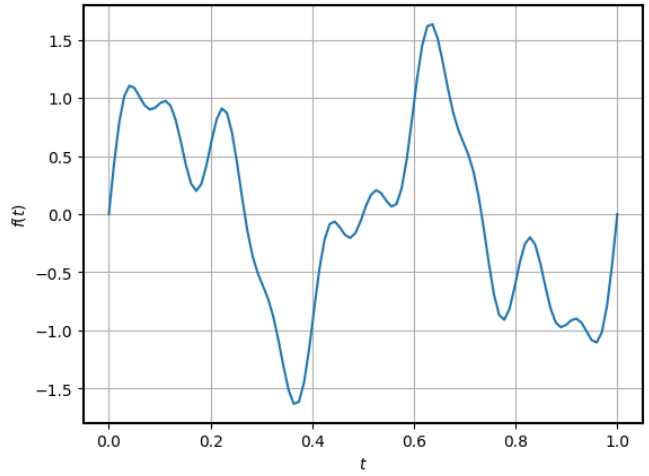


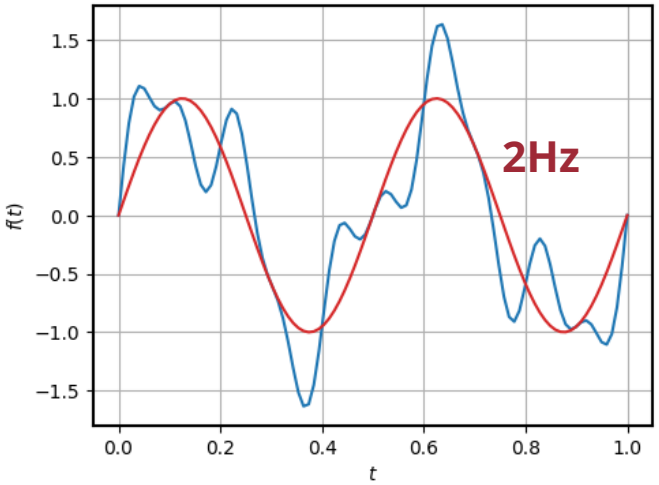
$$\sin(2 \cdot 2\pi t)$$

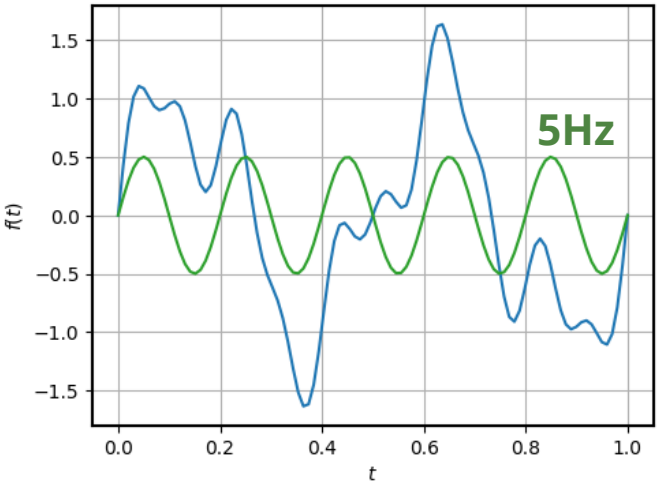


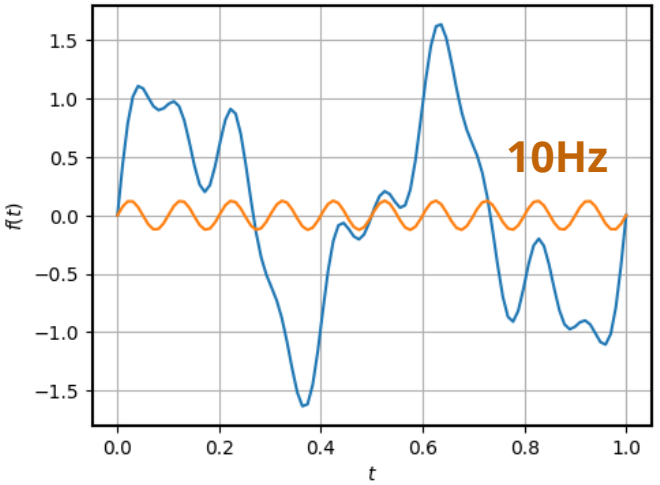
$$\frac{1}{2}\sin(10 \cdot 2\pi t)$$

Spectral Analysis



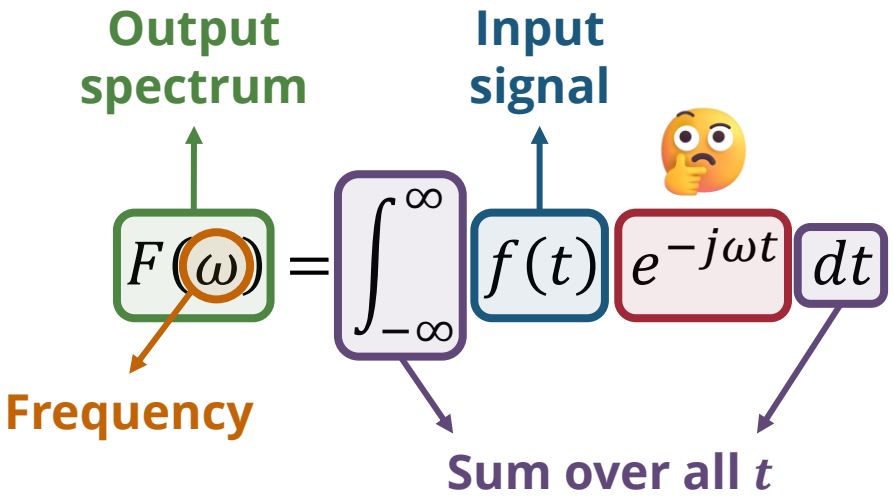






Fourier Transform

- **Intuition**: Decompose time-domain signals into **frequency components**
- Mathematical formulation:



Demystifying Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \boxed{e^{-j\omega t}} \, dt$$

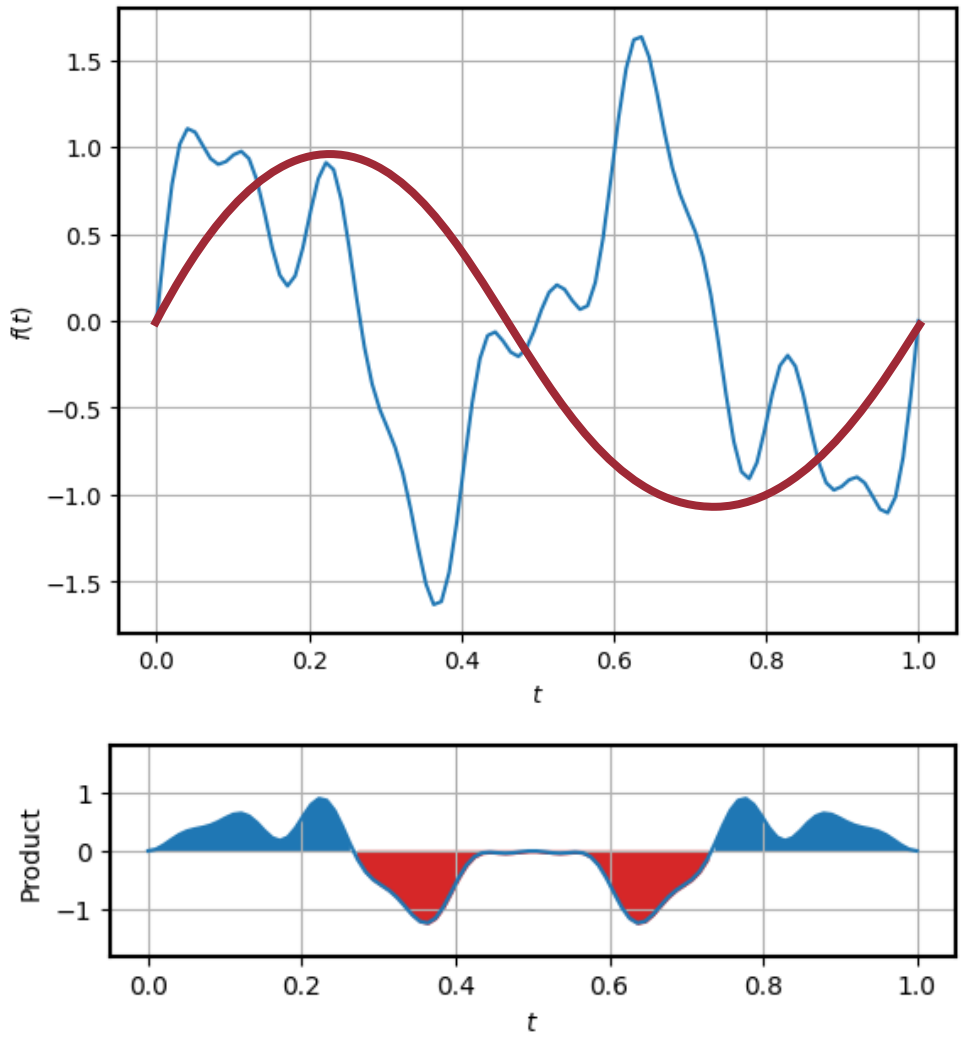
🤔

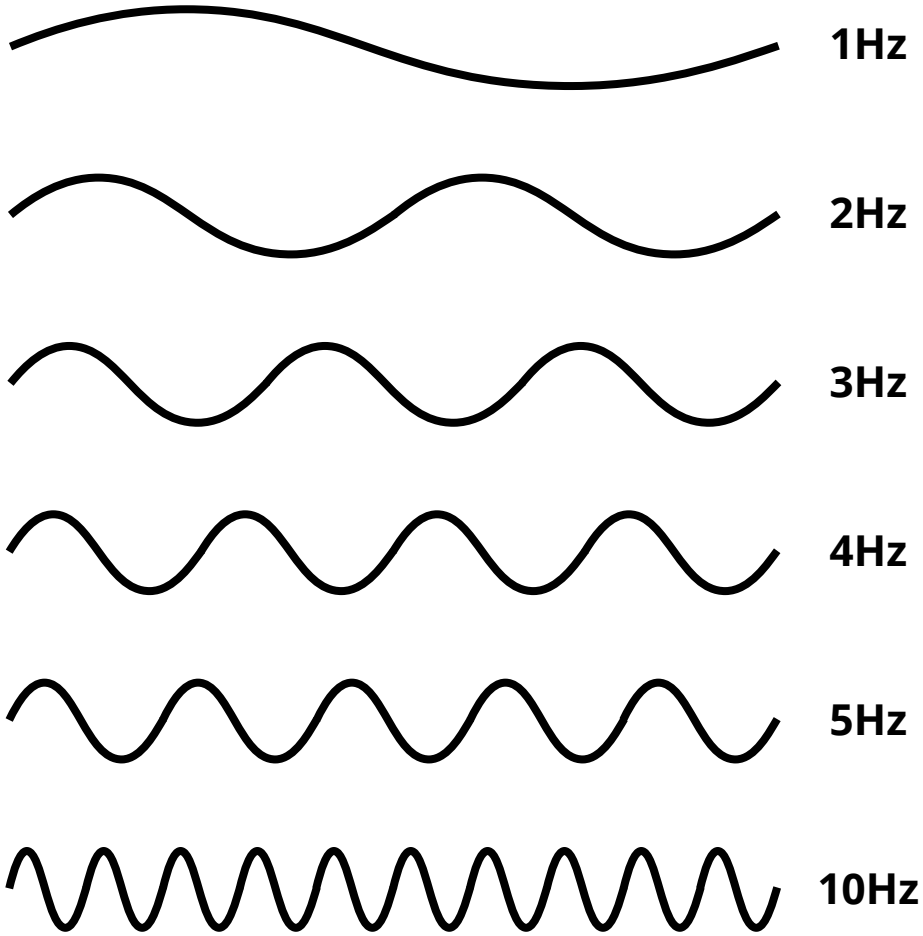
Euler's formula

$$e^{-j\theta} = \cos\theta + j\sin\theta$$

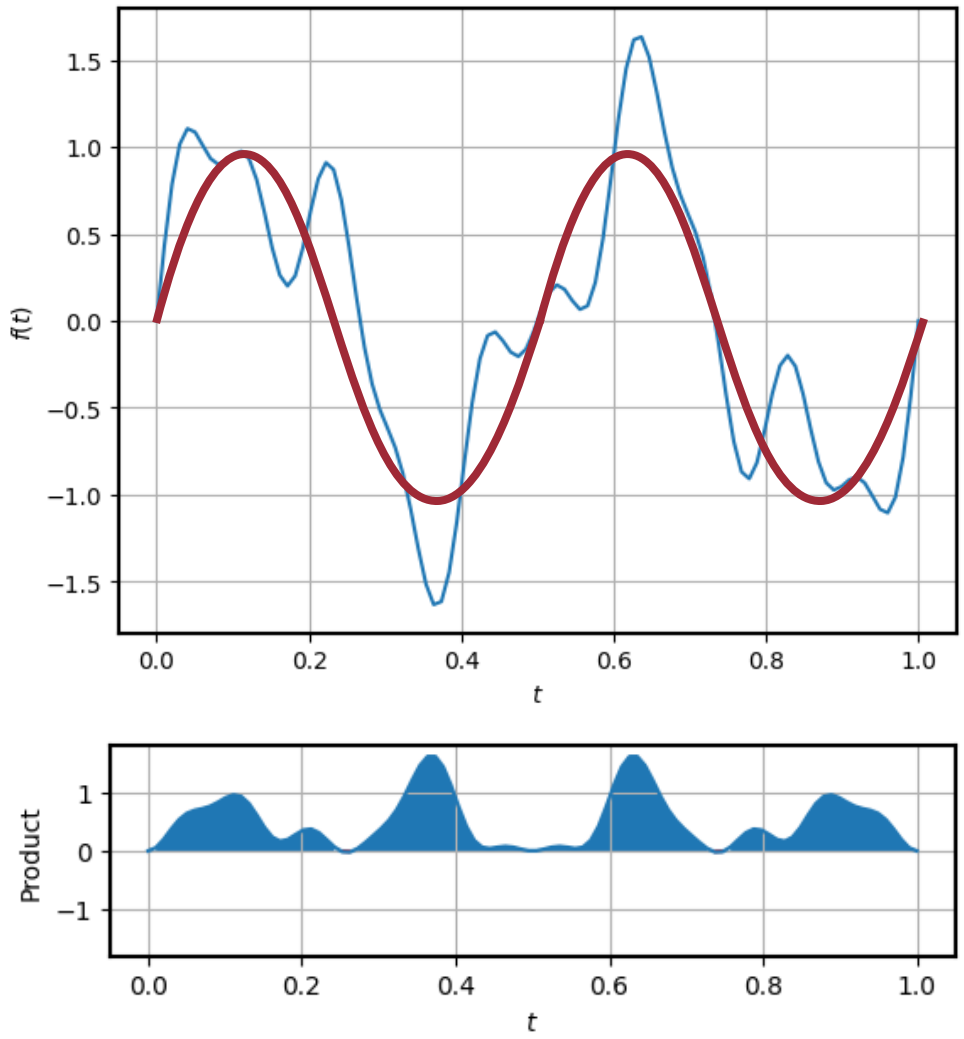
$$F(\omega) = \int_{-\infty}^{\infty} f(t)\boxed{\cos(-\omega t)} + j\, f(t)\boxed{\sin(-\omega t)} \, dt$$

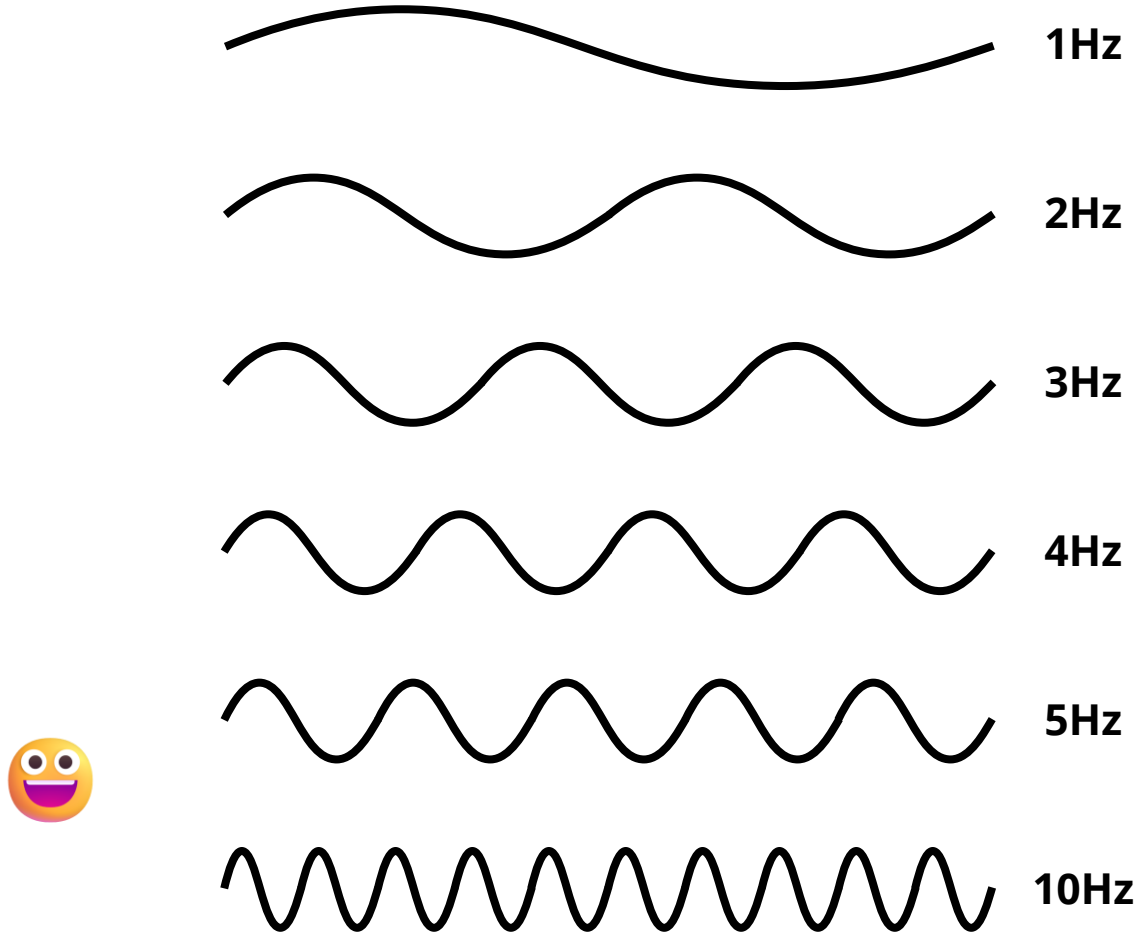
Demystifying Fourier Transform



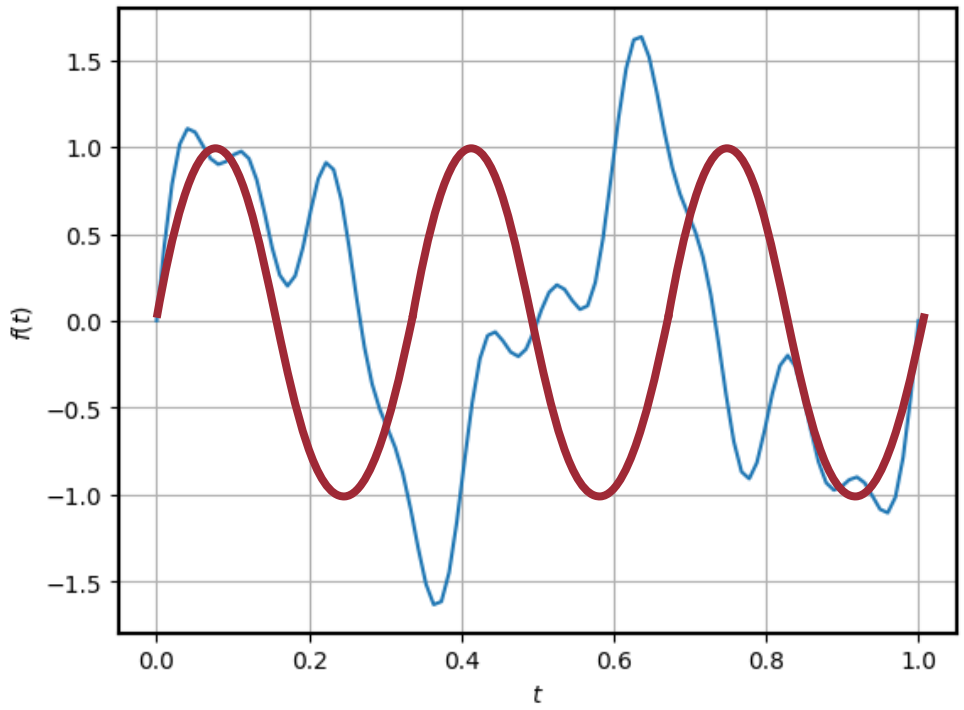


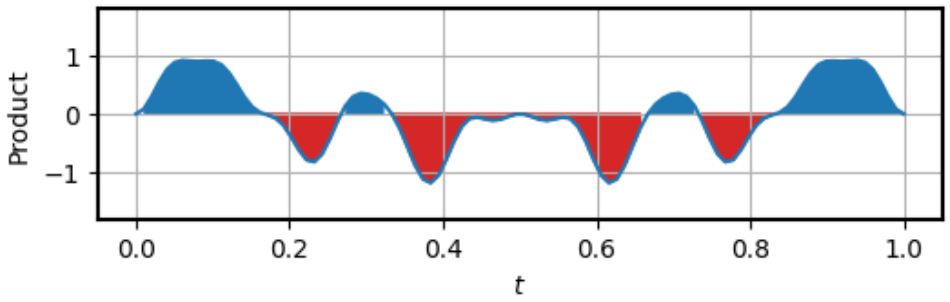
Demystifying Fourier Transform

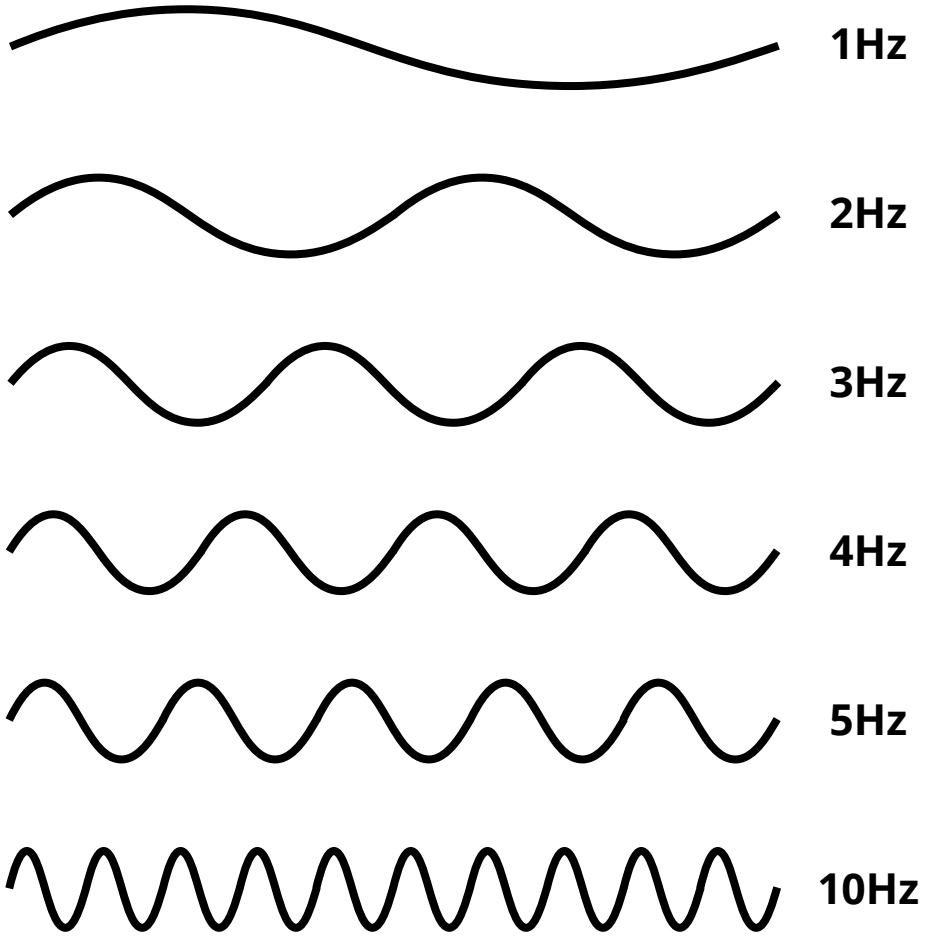




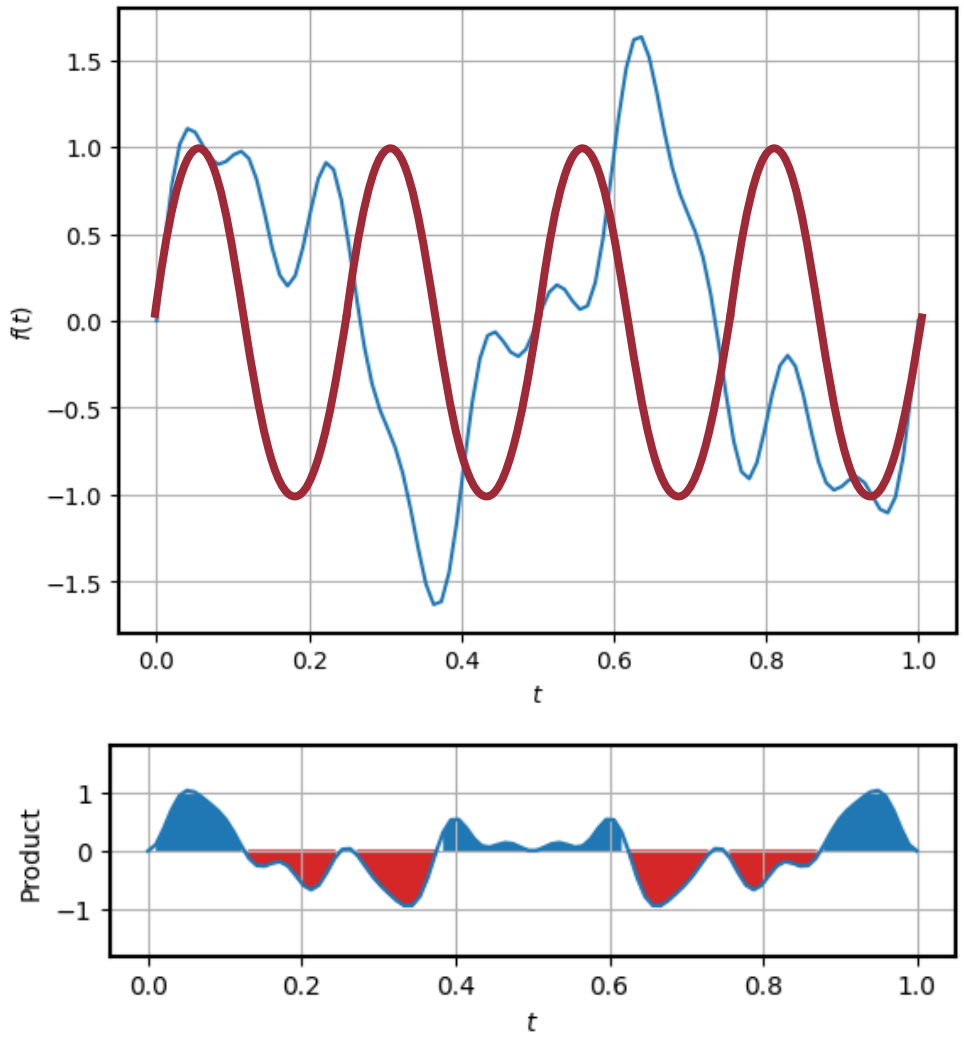
Demystifying Fourier Transform

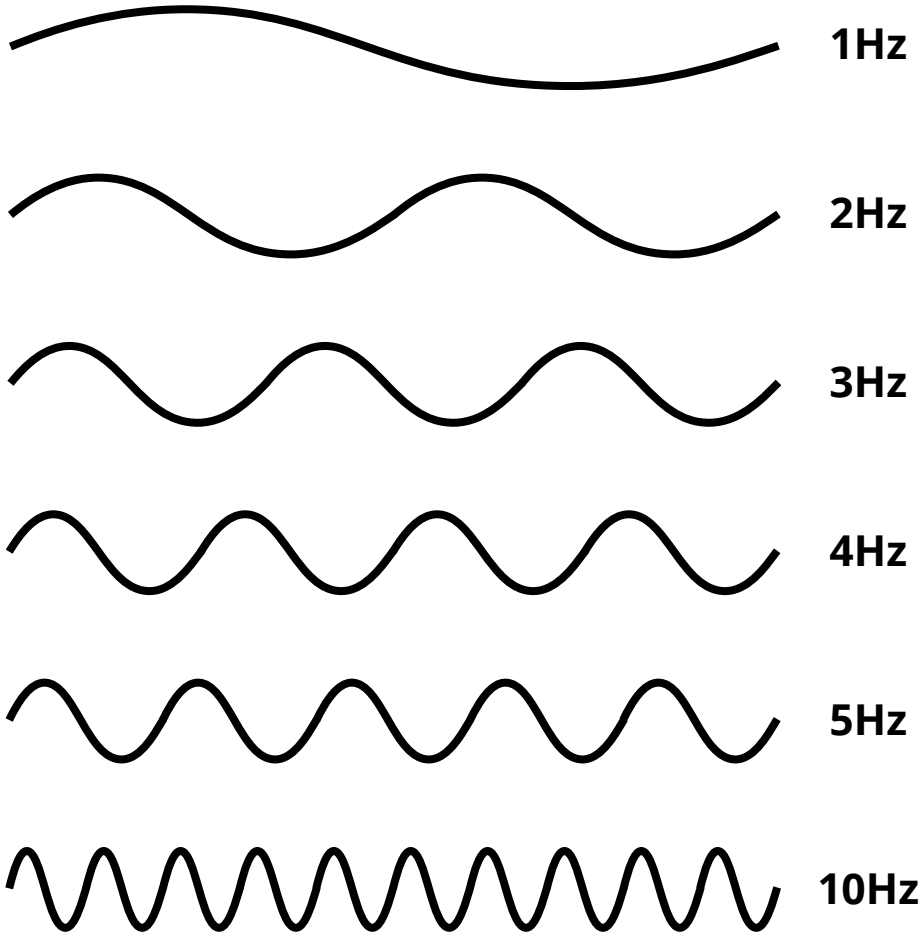




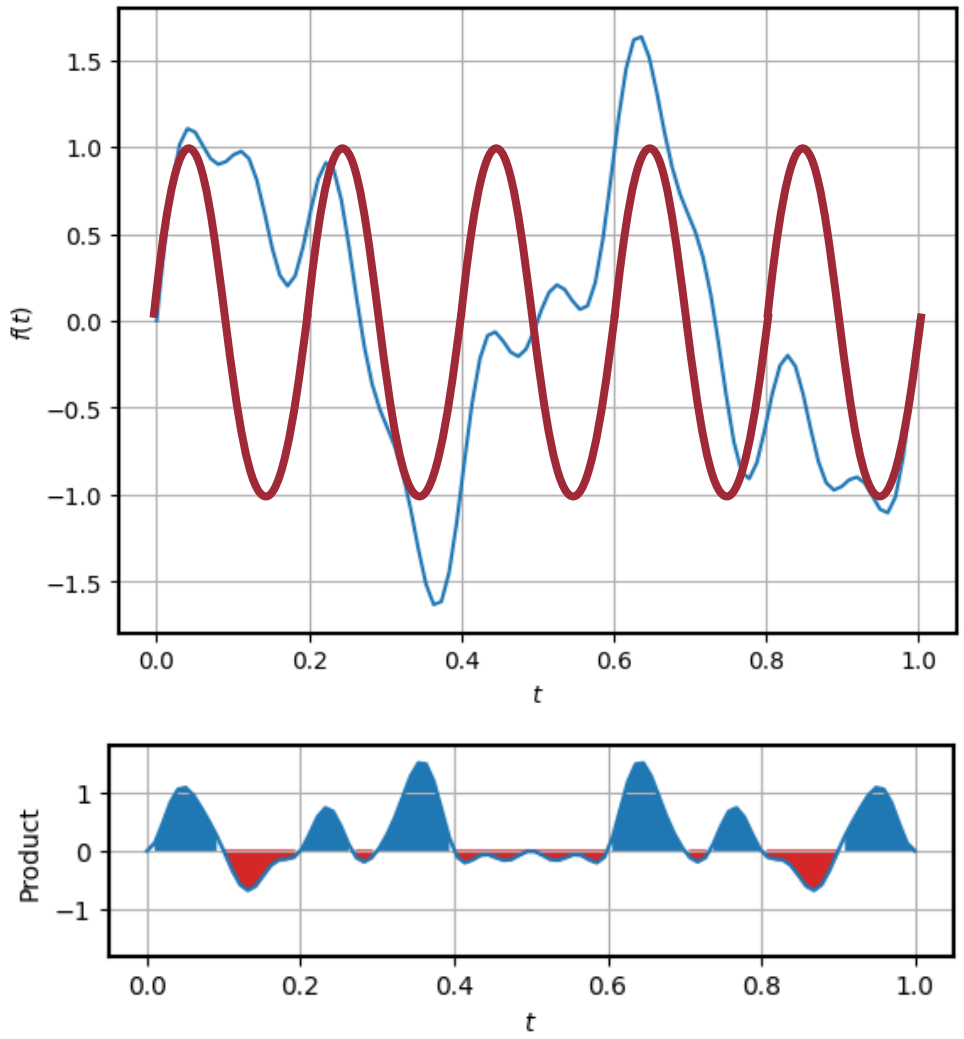


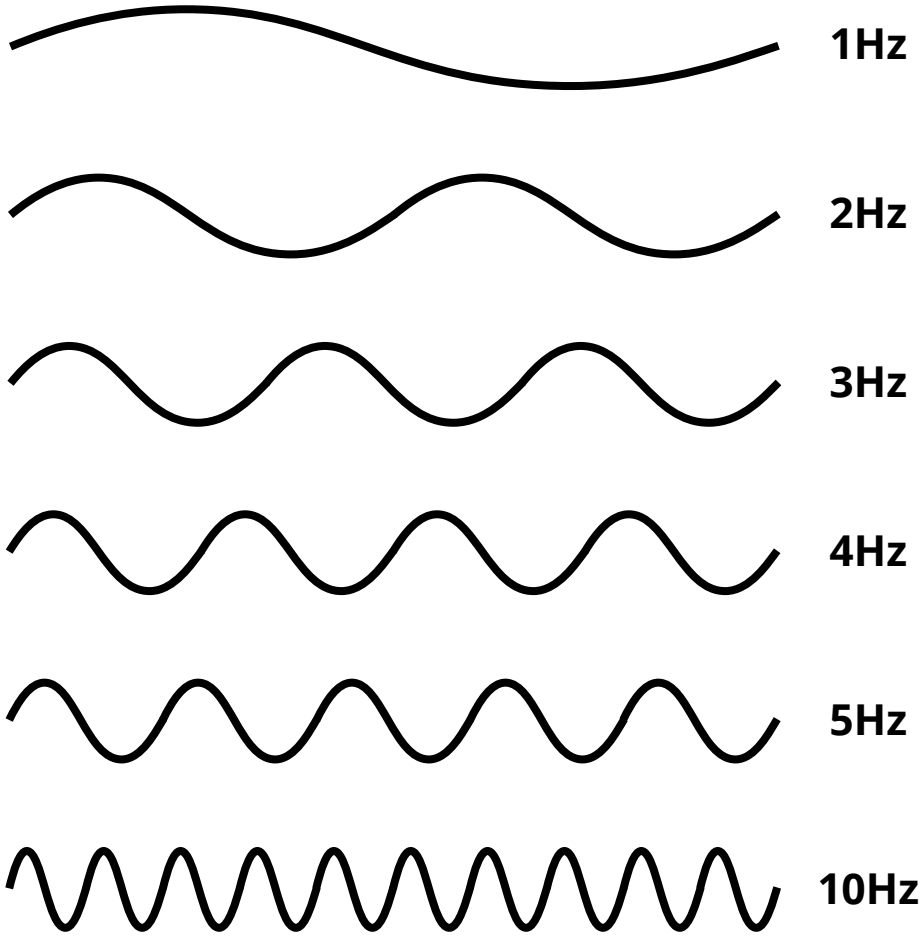
Demystifying Fourier Transform



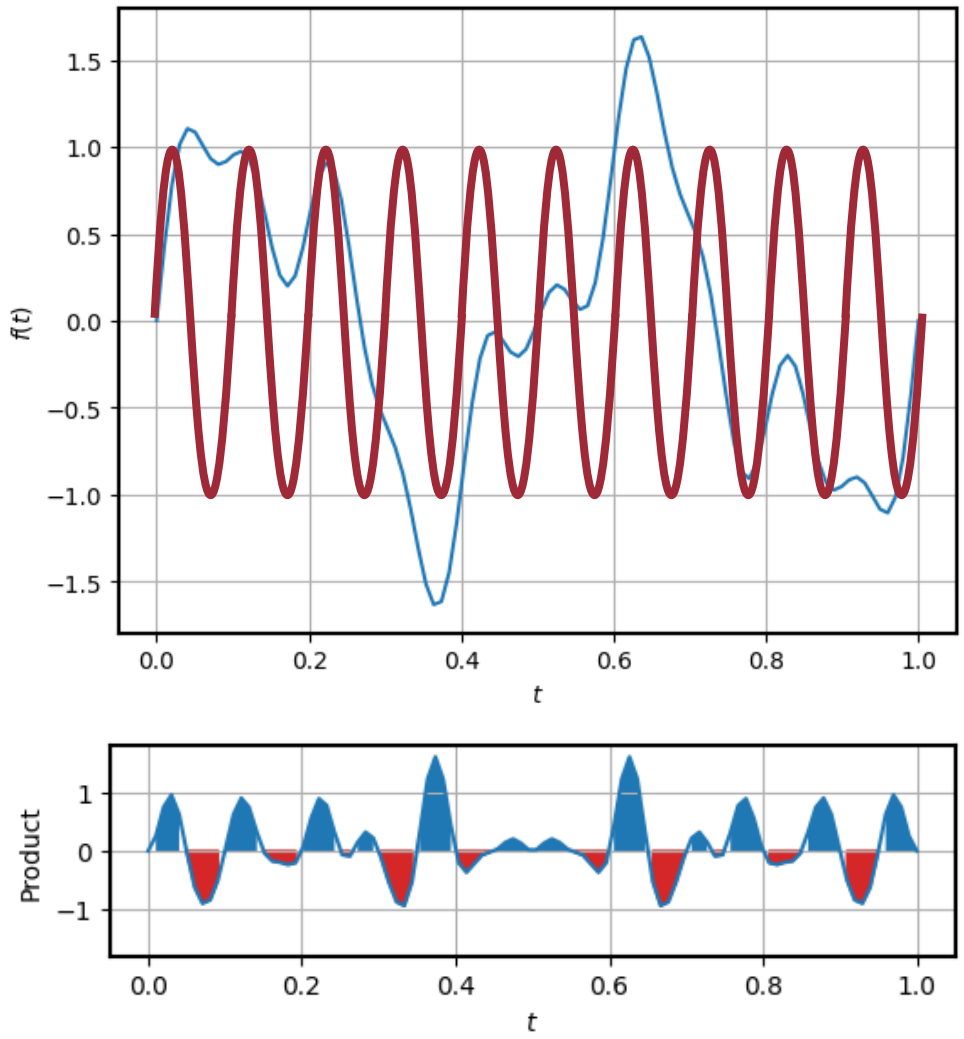


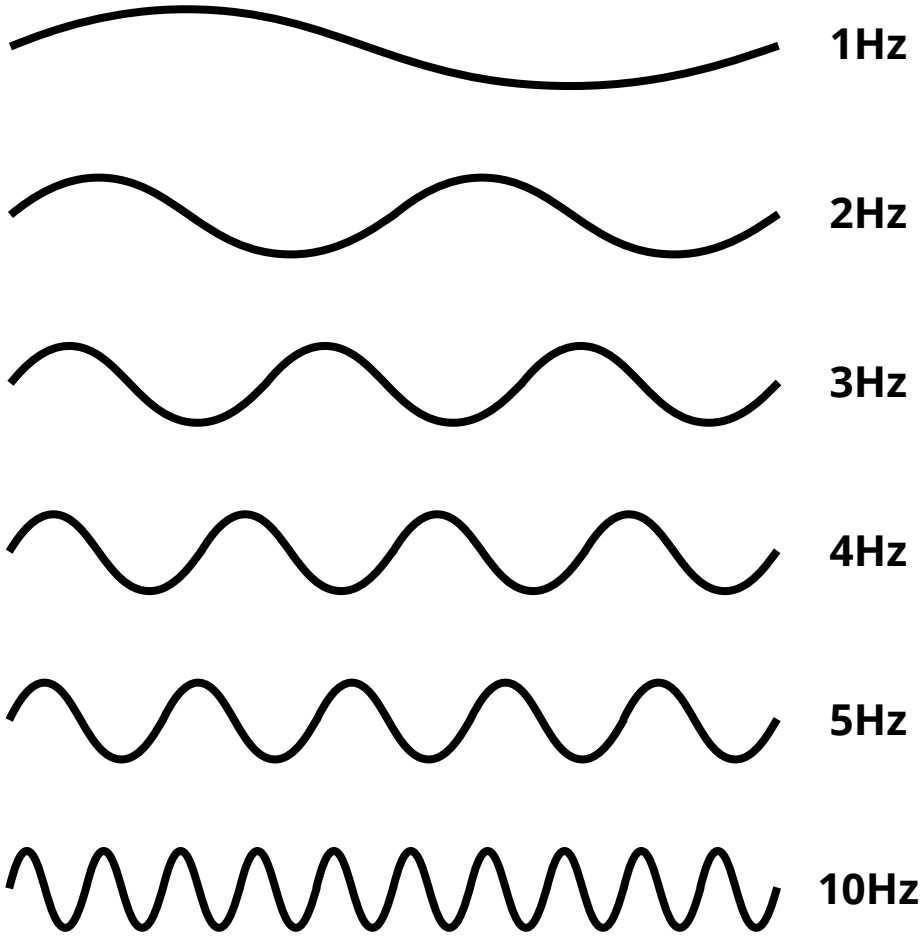
Demystifying Fourier Transform





Demystifying Fourier Transform

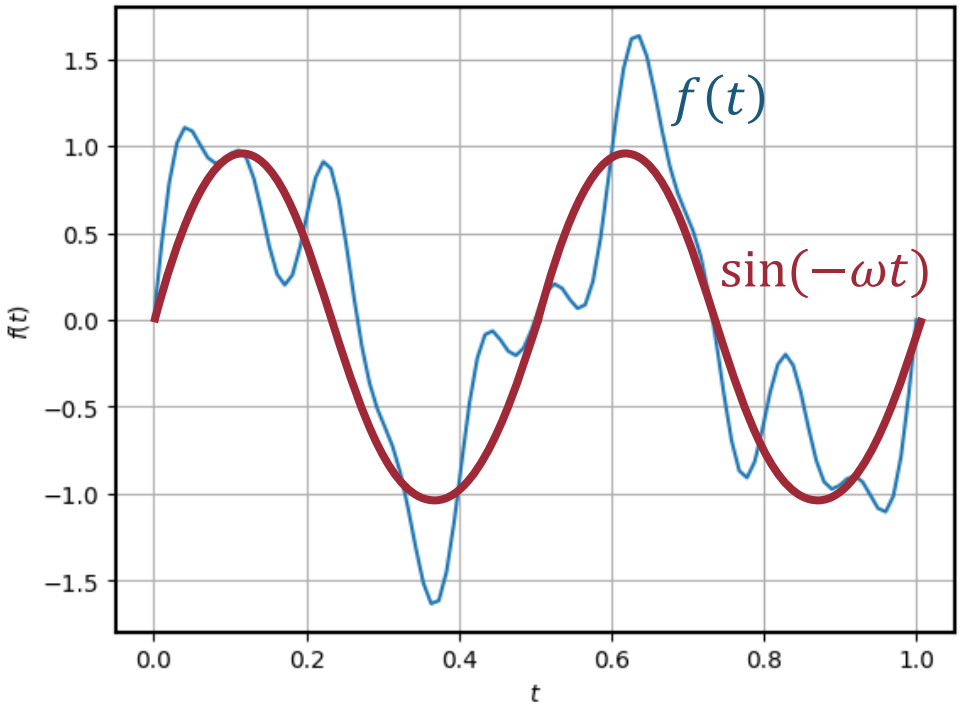




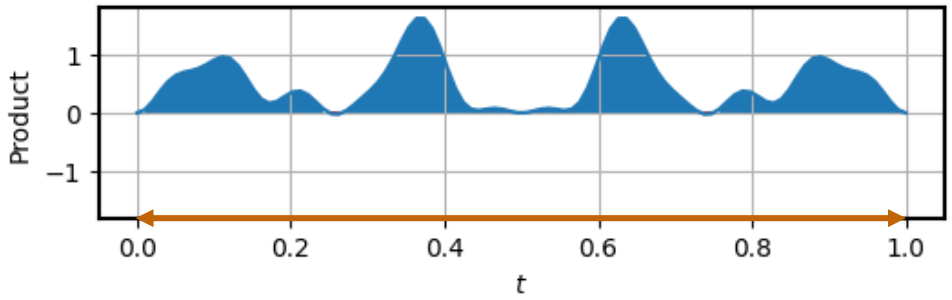
Demystifying Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\cos(-\omega t) + j\, f(t)\sin(-\omega t)\, dt$$

Sum over all t



$f(t)\sin(-\omega t)$

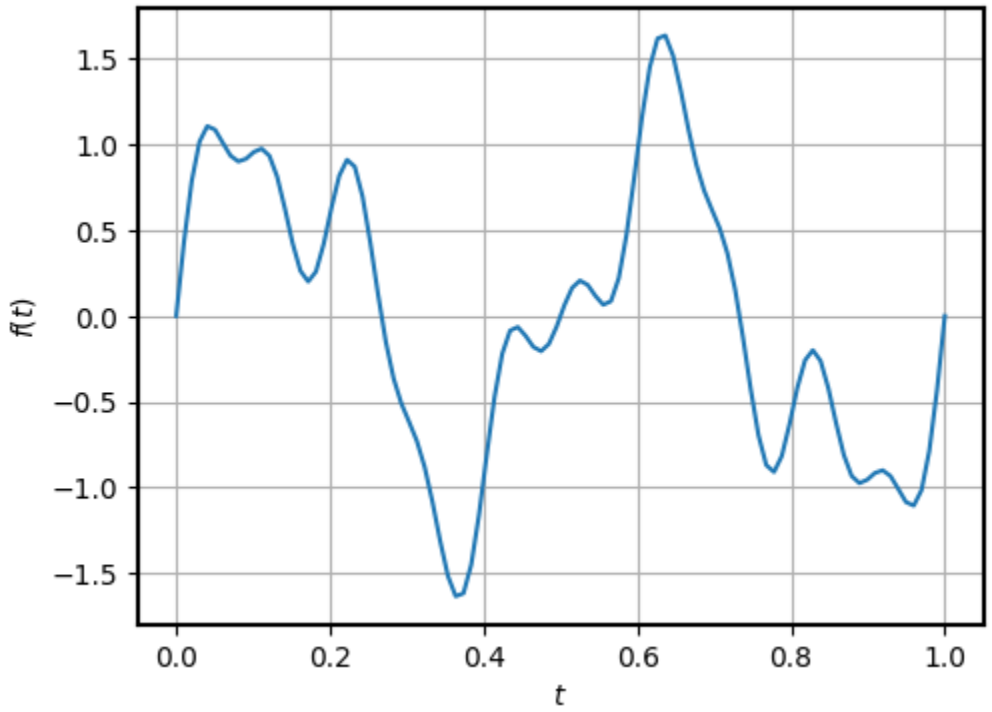


Sum = 0.495

Demystifying Fourier Transform

Signal

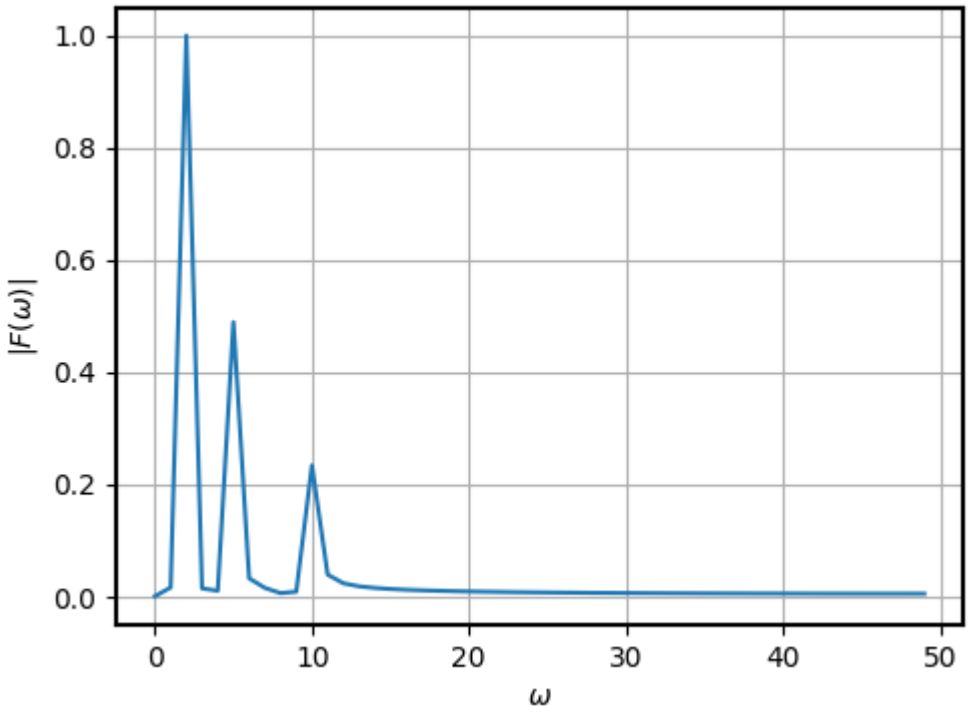
(time-domain)



Fourier Transform

Spectrum

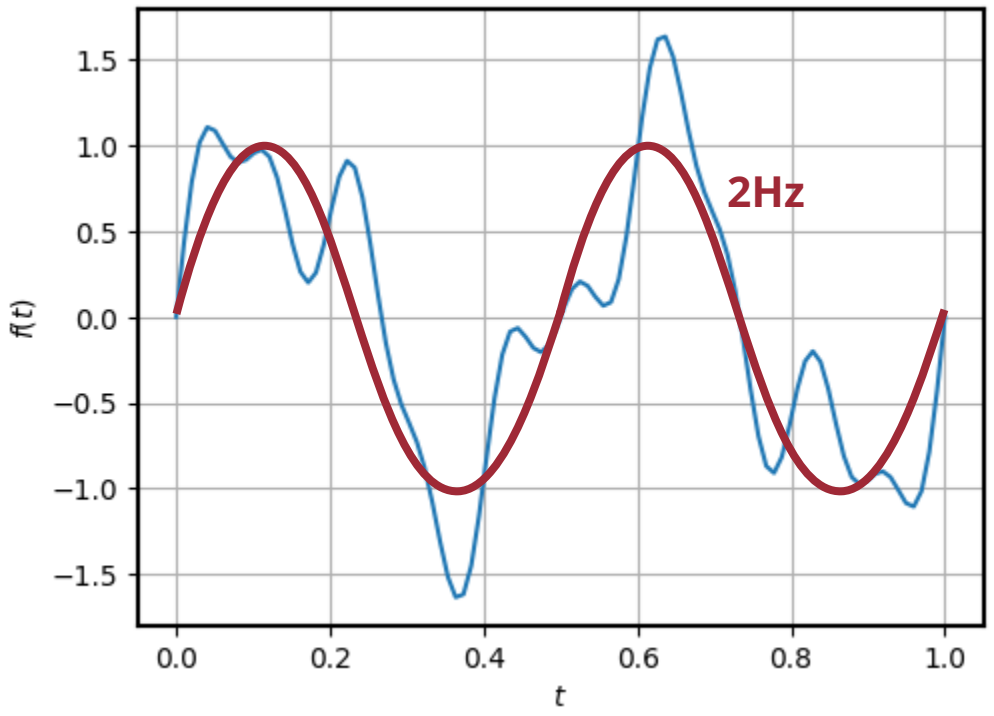
(frequency-domain)



Demystifying Fourier Transform

Signal

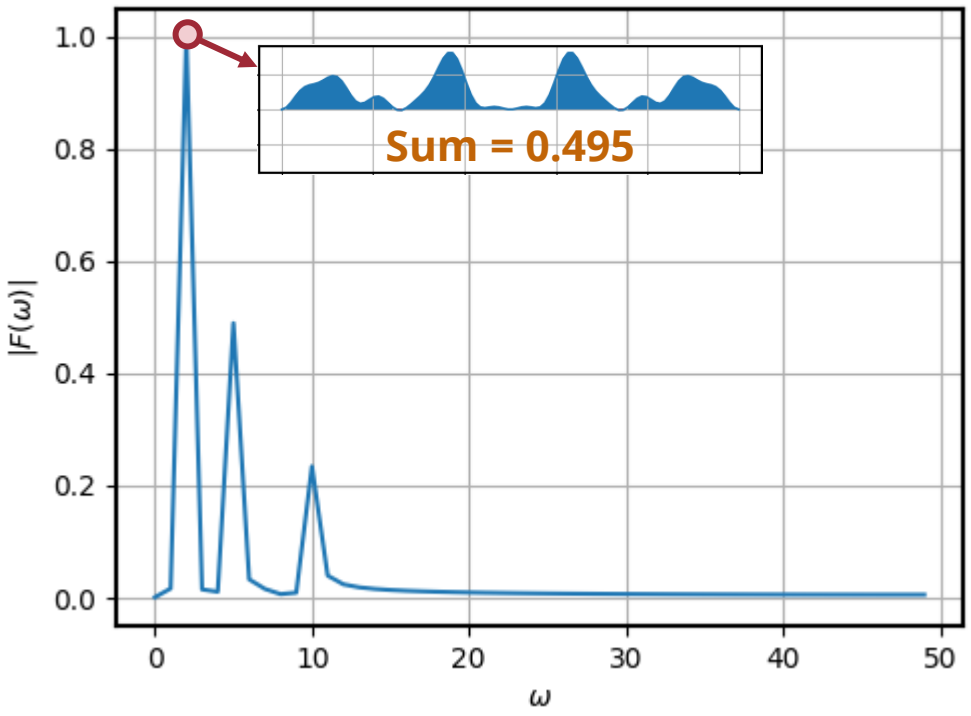
(time-domain)



Fourier Transform

Spectrum

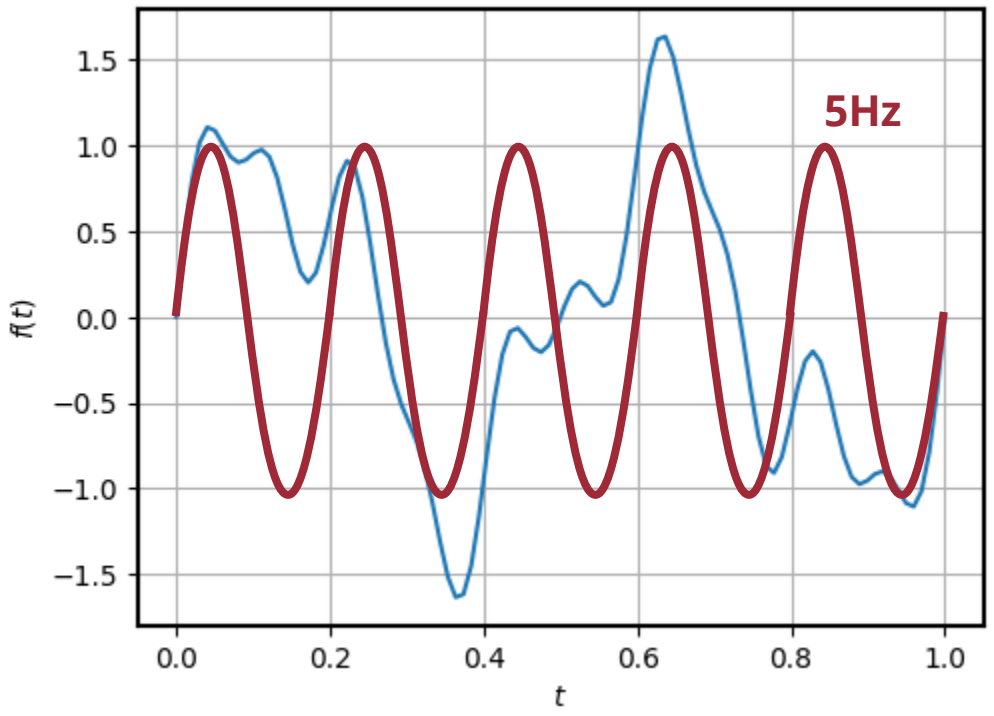
(frequency-domain)



Demystifying Fourier Transform

Signal

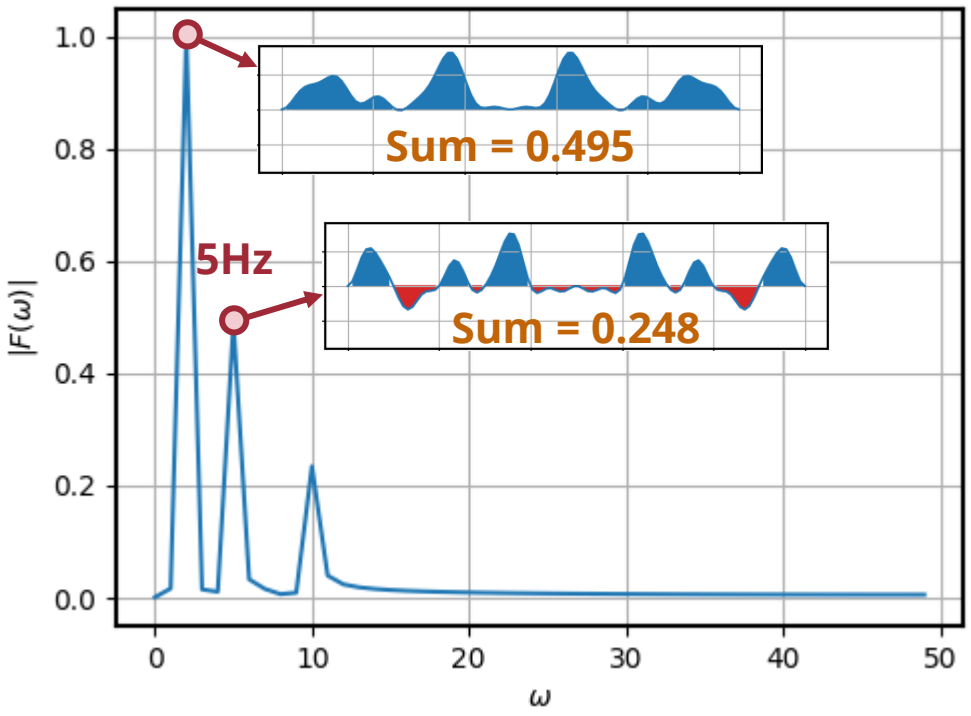
(time-domain)



Fourier Transform

Spectrum

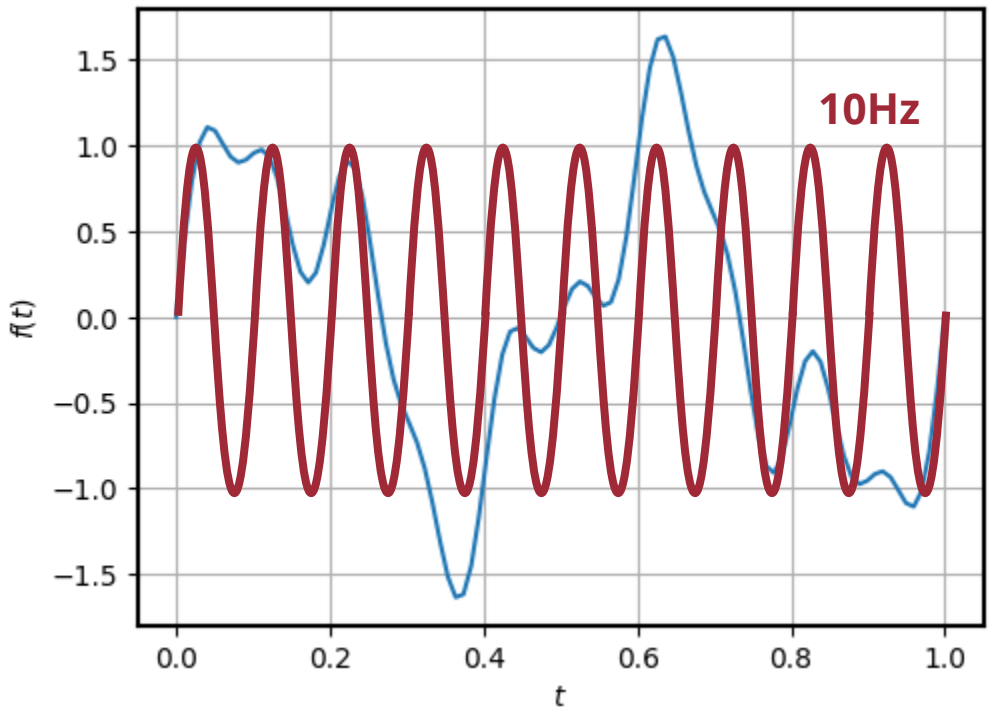
(frequency-domain)



Demystifying Fourier Transform

Signal

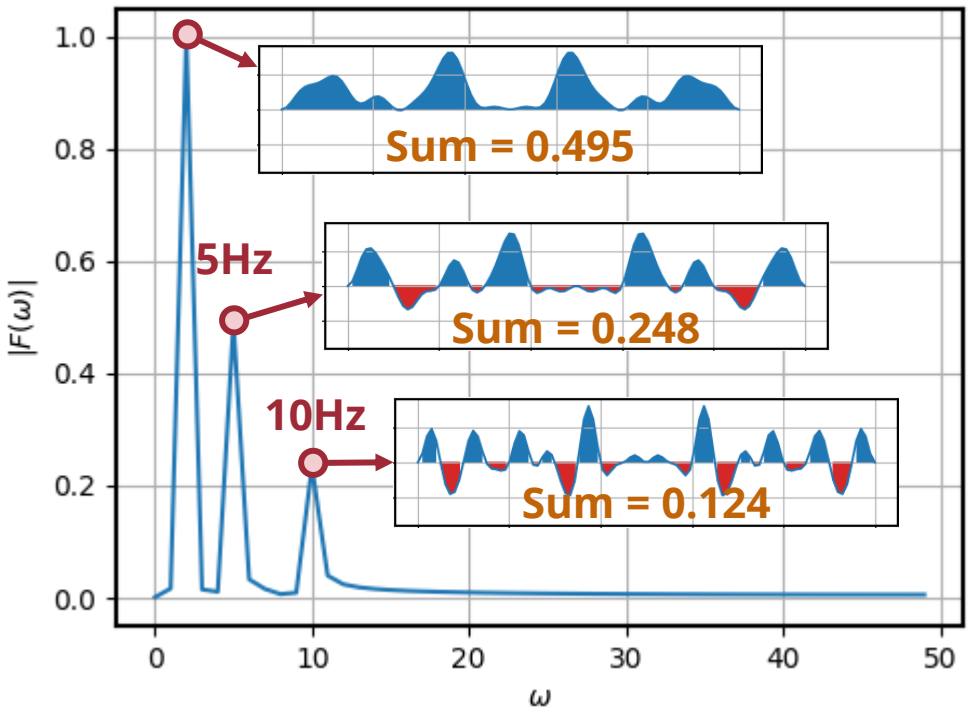
(time-domain)



Fourier Transform

Spectrum

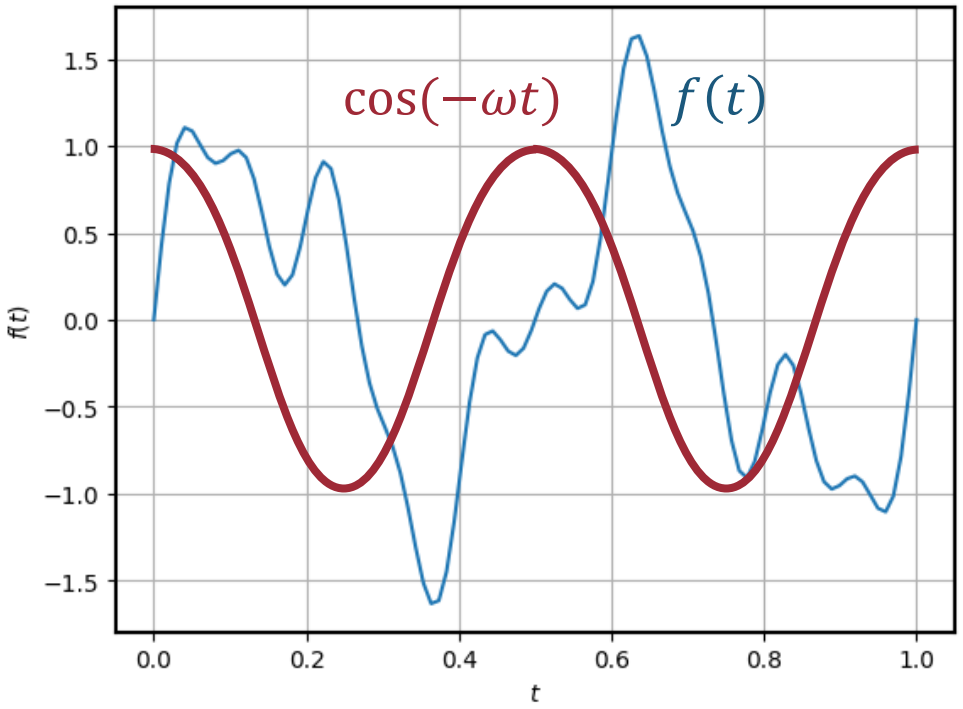
(frequency-domain)



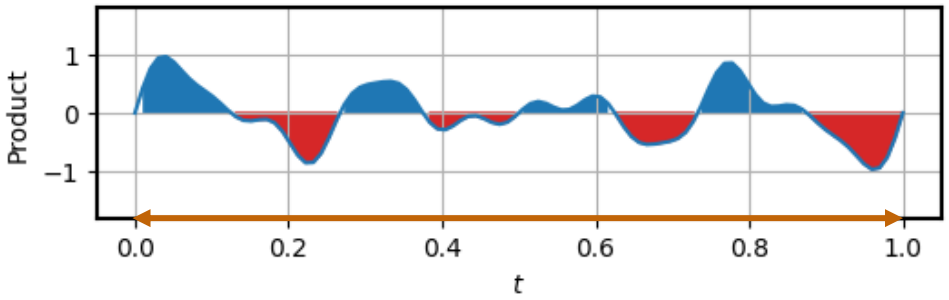
Demystifying Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\cos(-\omega t) + j\, f(t)\sin(-\omega t)\, dt$$

Sum over all t

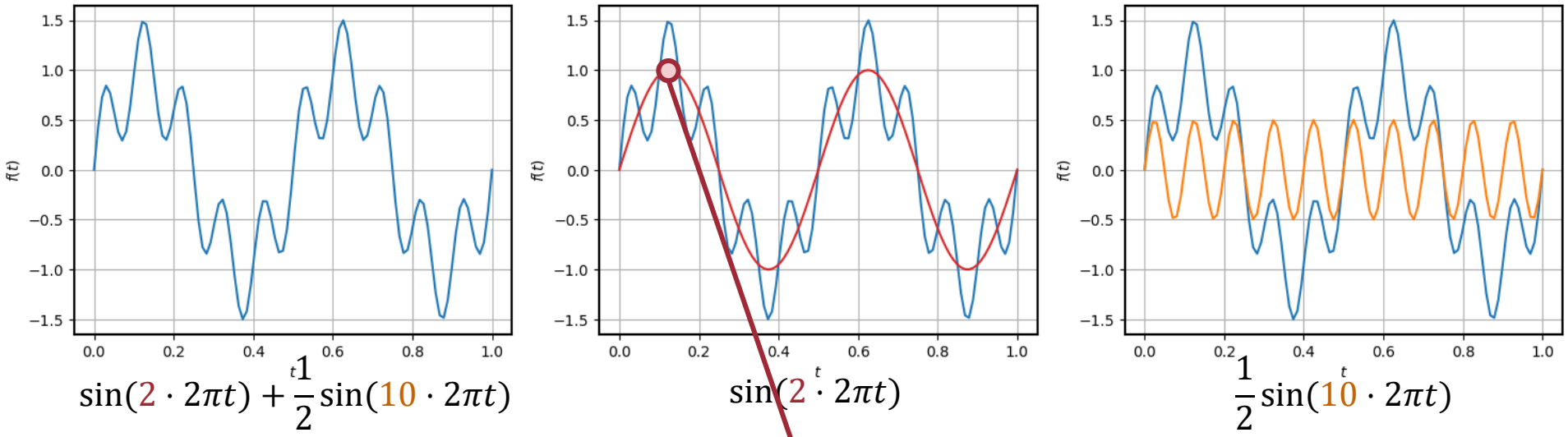


$f(t)\cos(-\omega t)$

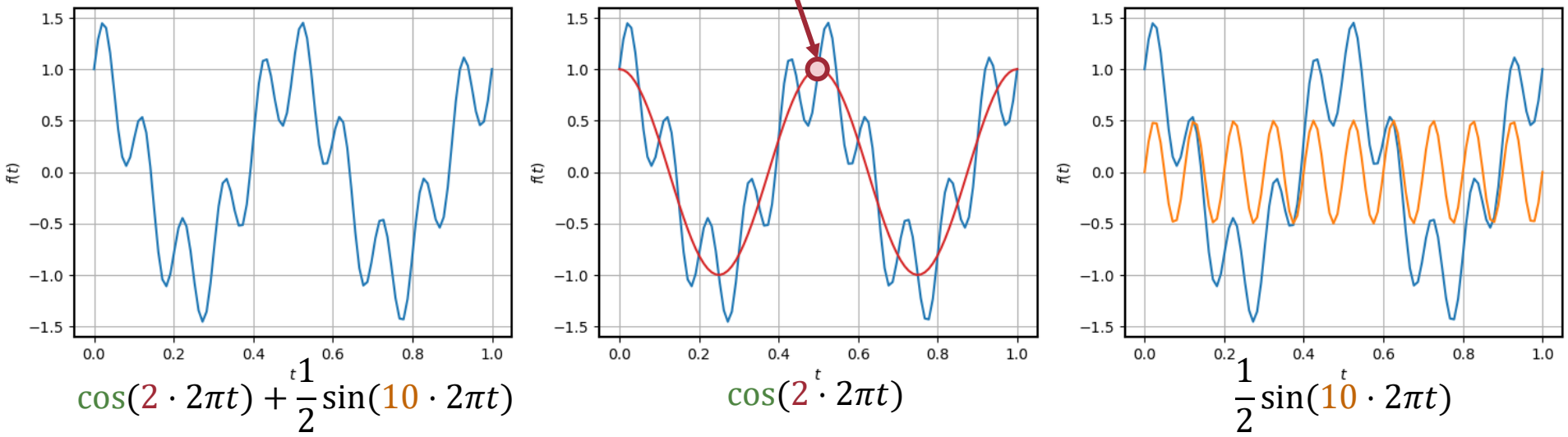


Sum = 0

Phase



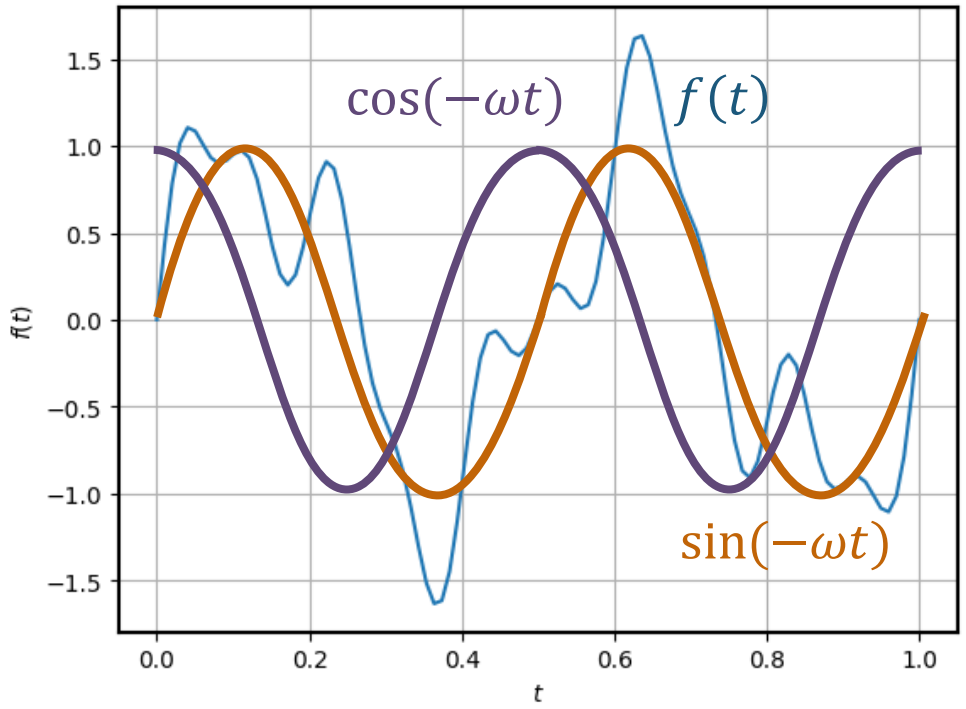
$\sin(2 \cdot 2\pi t) + \frac{1}{2}\sin(10 \cdot 2\pi t)$ $\quad \sin(2 \cdot 2\pi t) \quad \frac{1}{2}\sin(10 \cdot 2\pi t)$



$\cos(2 \cdot 2\pi t) + \frac{1}{2}\sin(10 \cdot 2\pi t)$ $\quad \cos(2 \cdot 2\pi t) \quad \frac{1}{2}\sin(10 \cdot 2\pi t)$

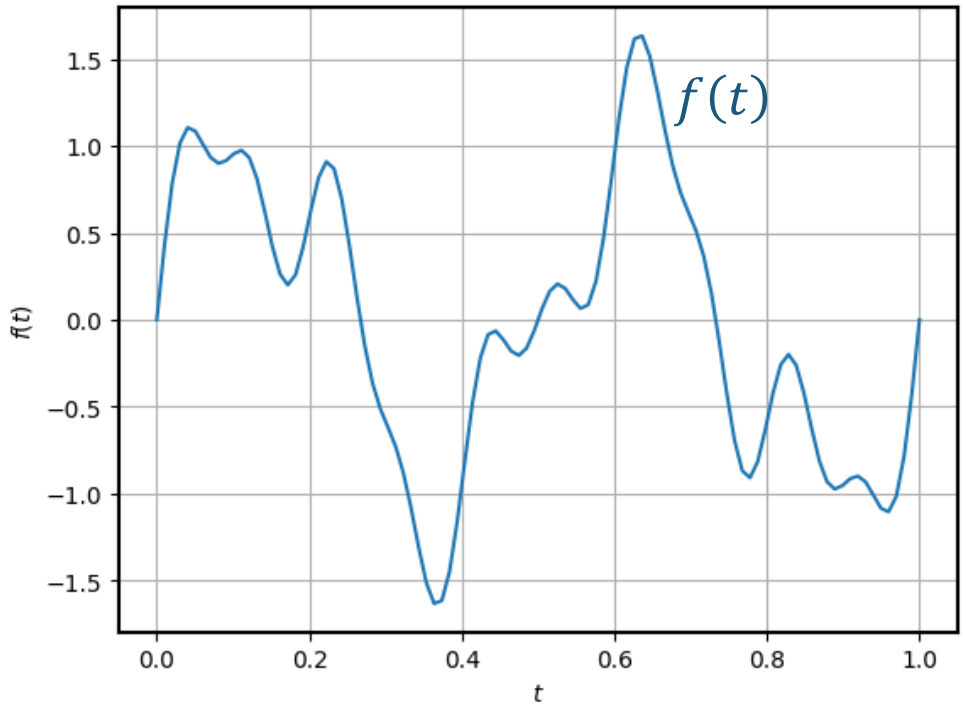
Demystifying Fourier Transform

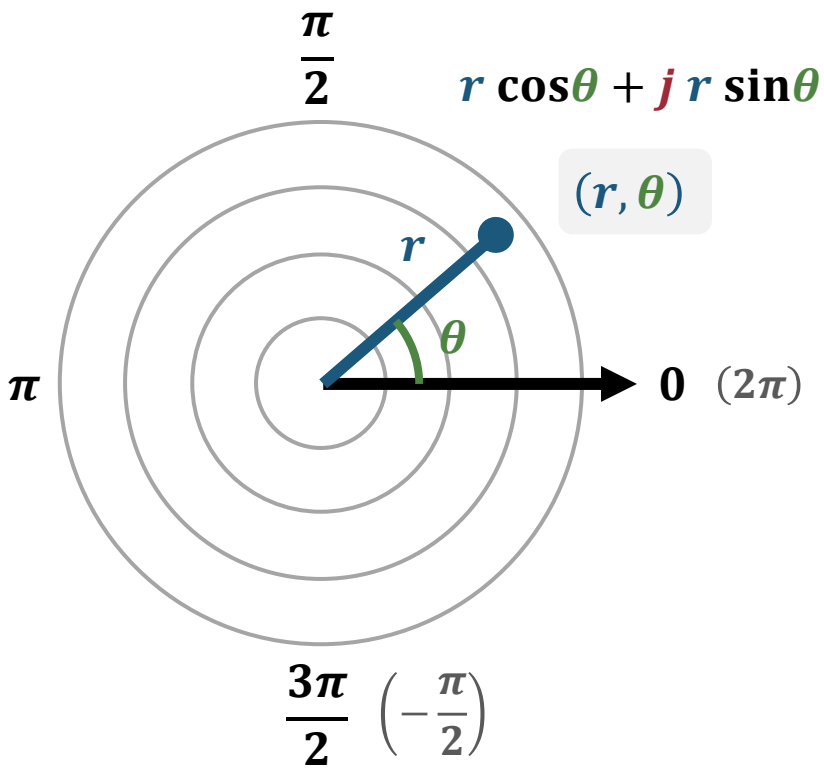
$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$



Demystifying Fourier Transform

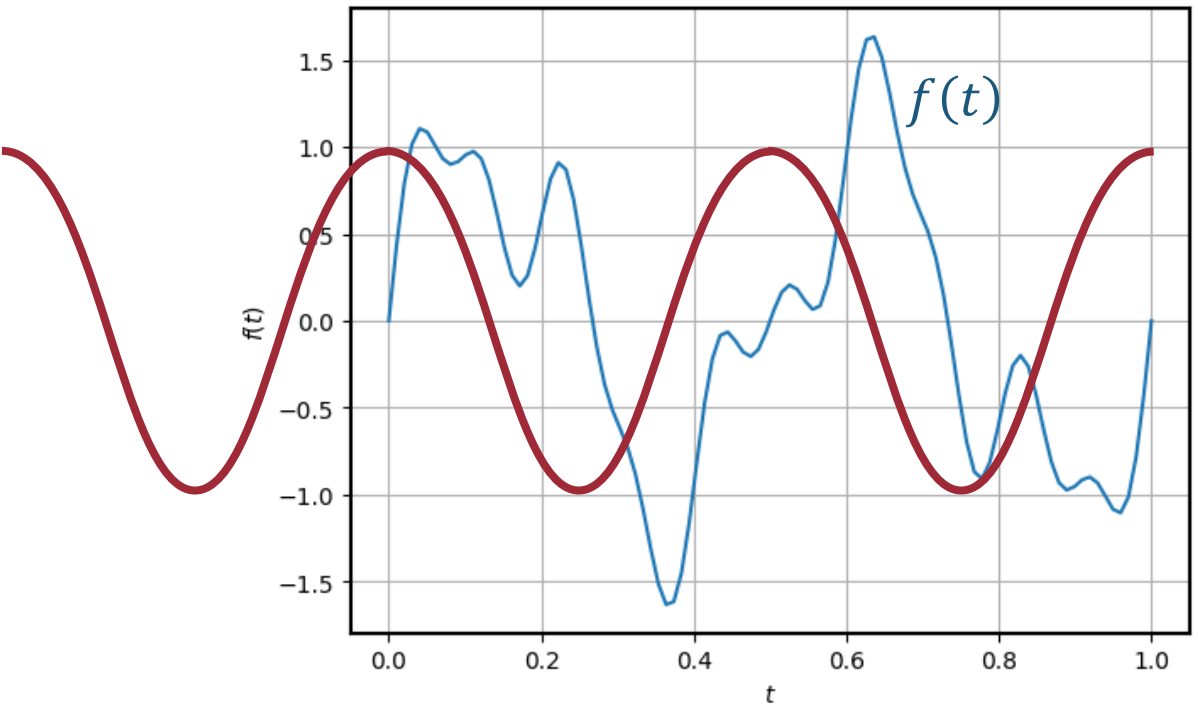
$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$

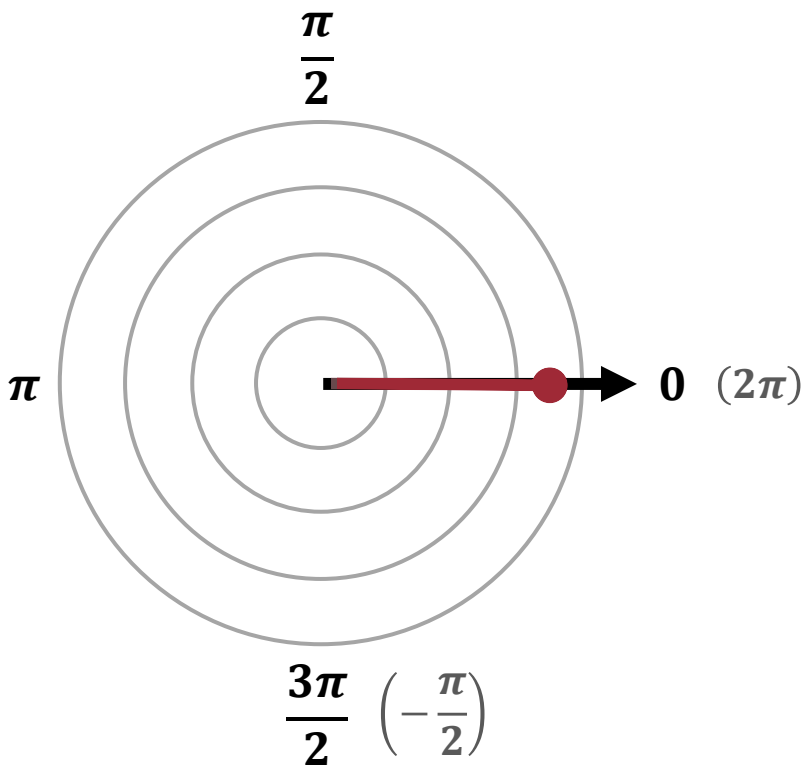




Demystifying Fourier Transform

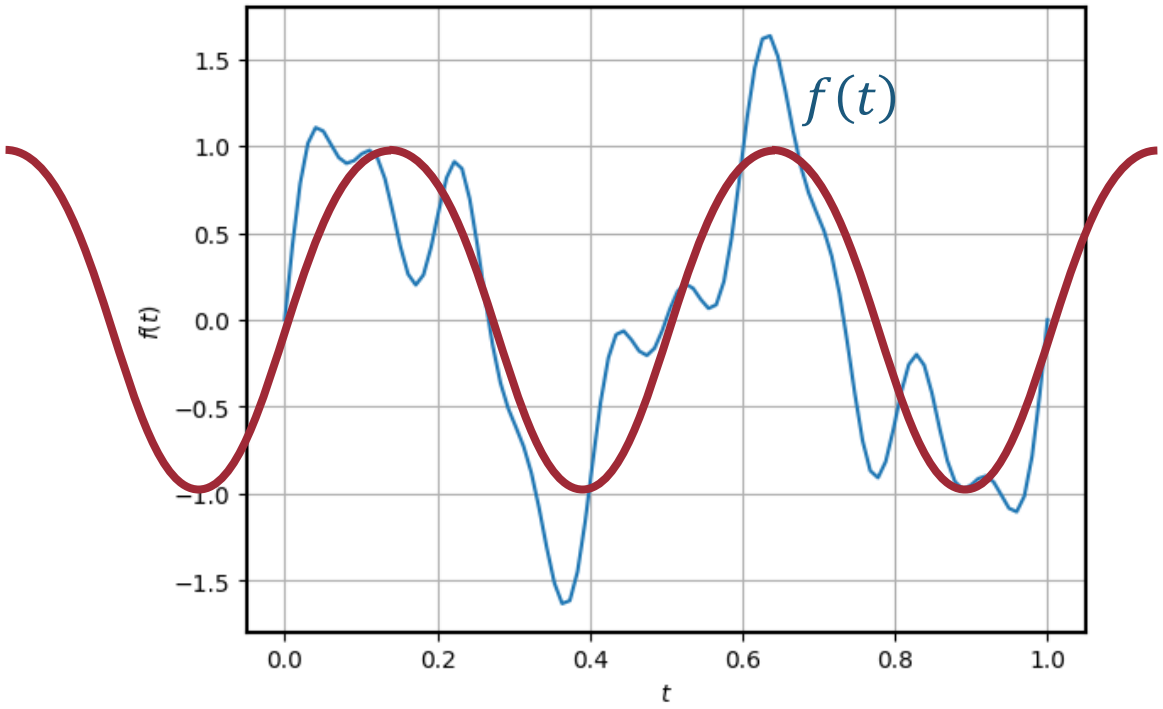
$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$

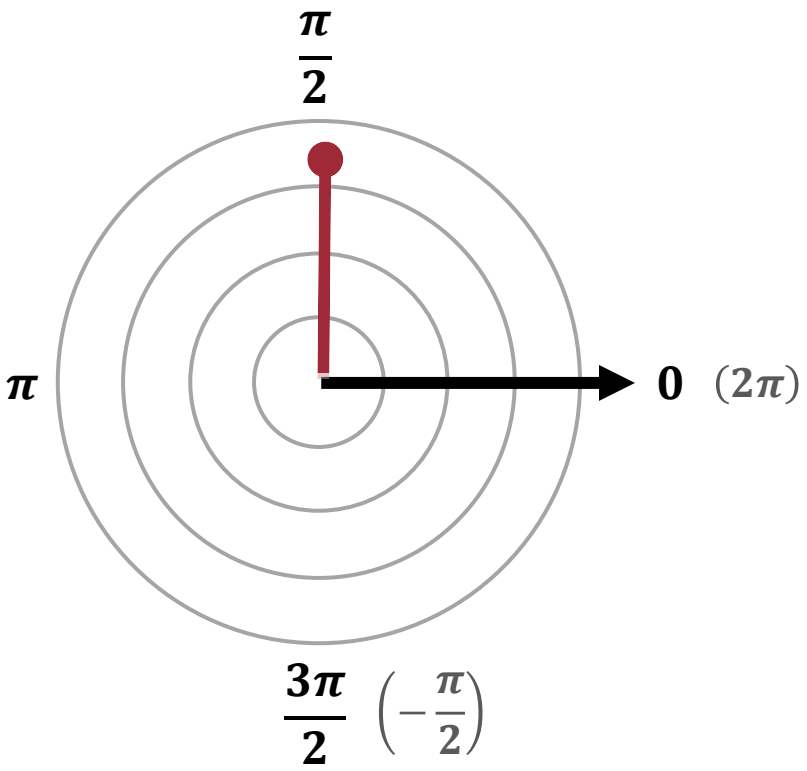




Demystifying Fourier Transform

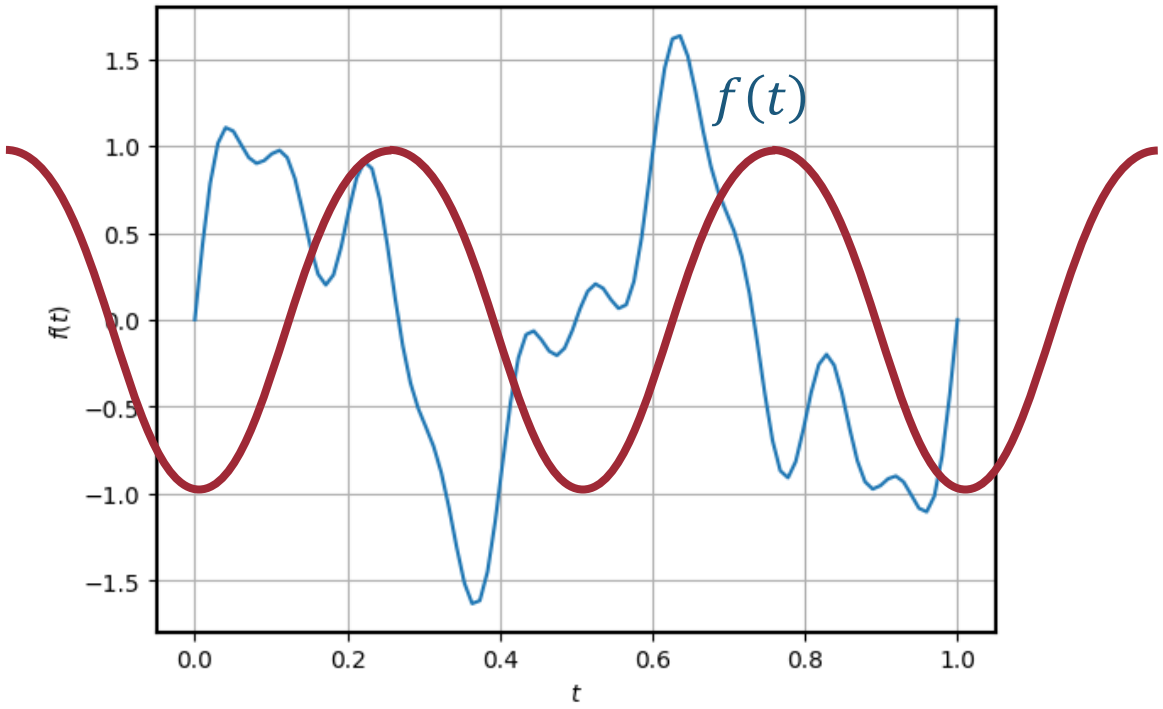
$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$

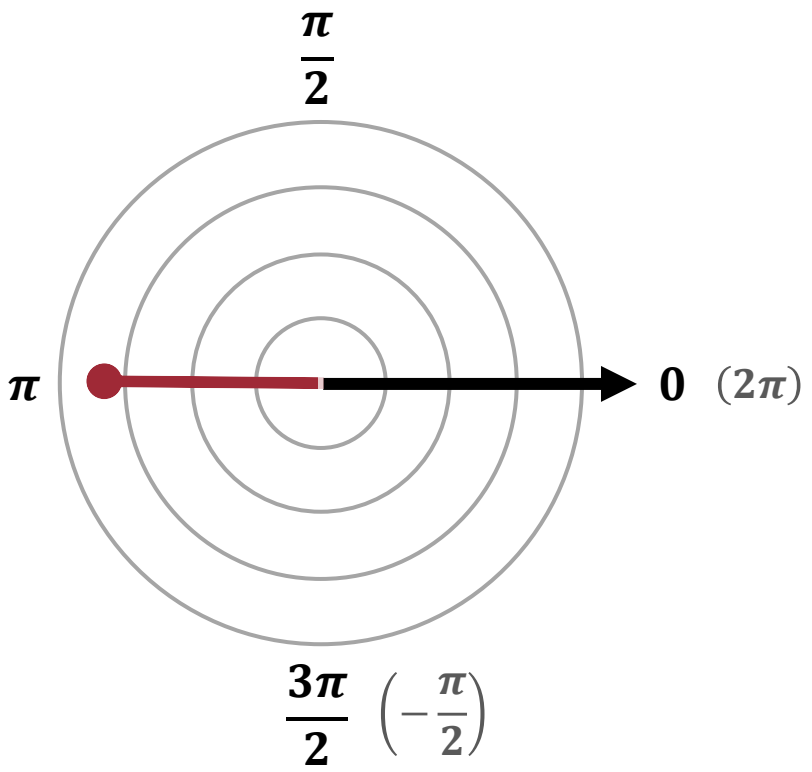




Demystifying Fourier Transform

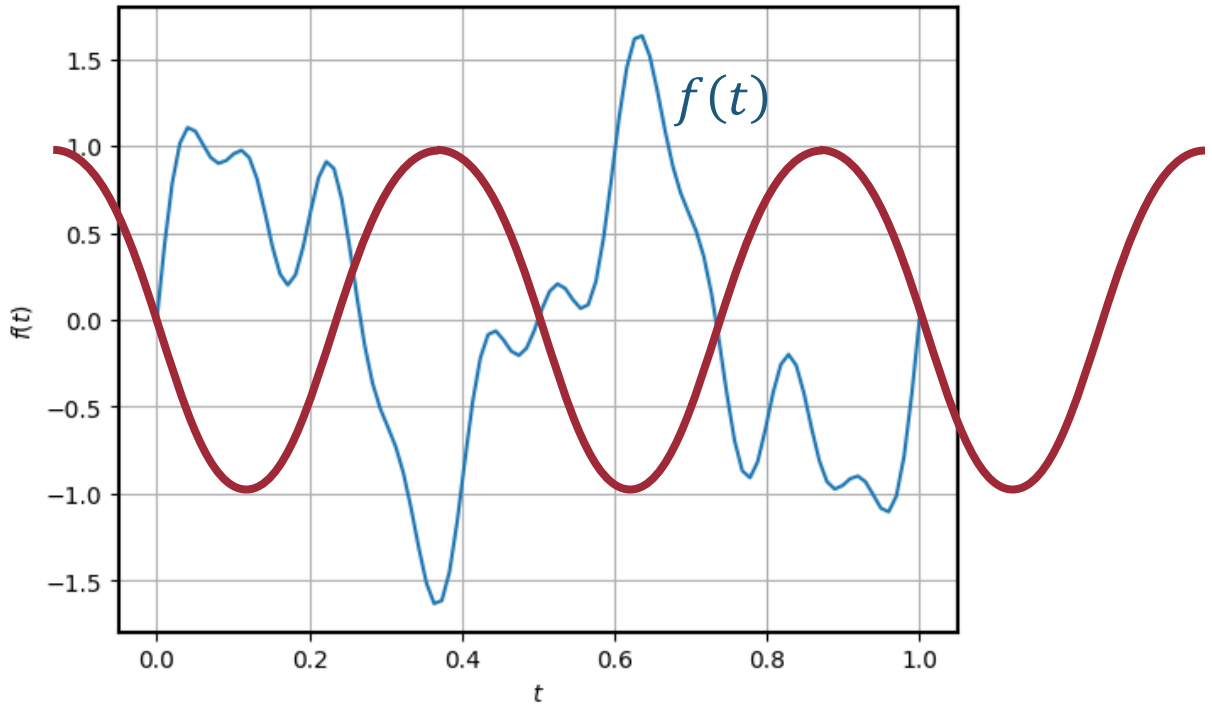
$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$

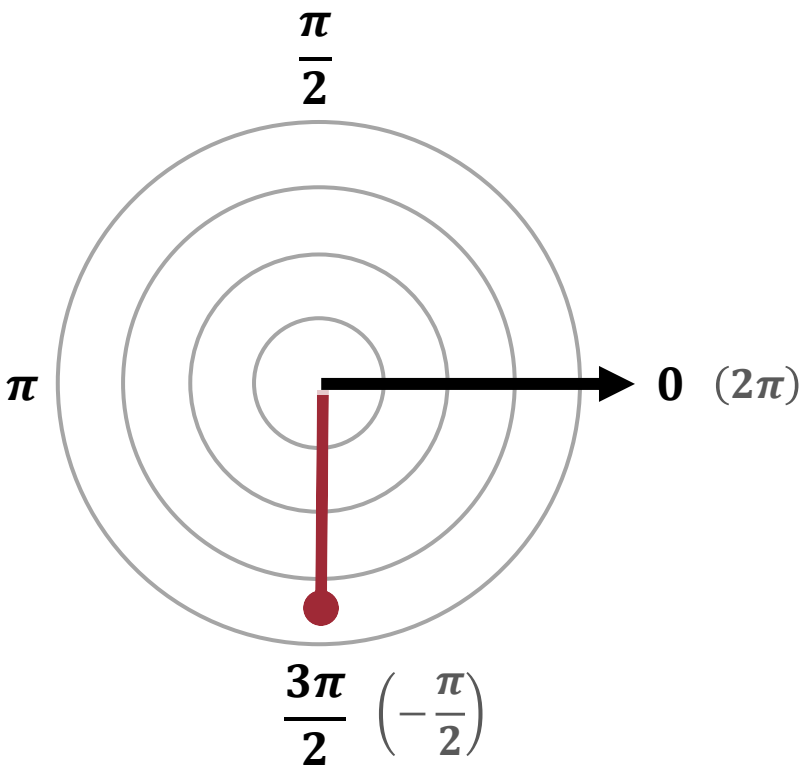




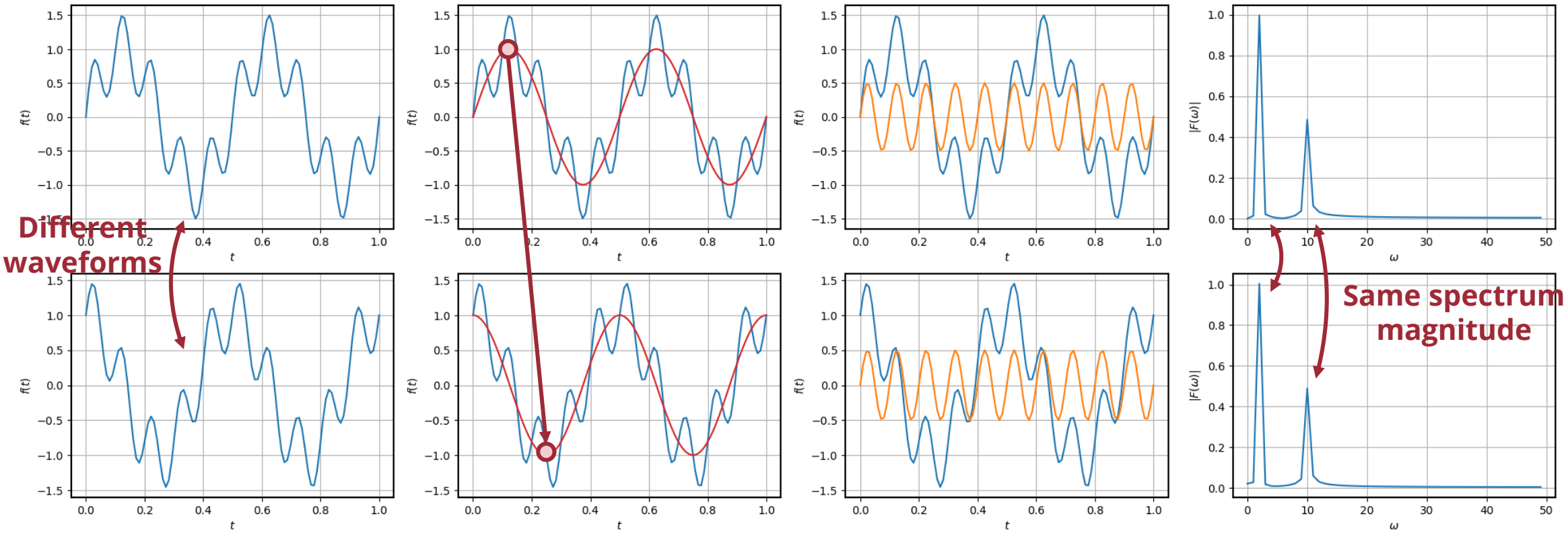
Demystifying Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)\cos(-\omega t)}_{\text{Real part}} + \underbrace{j f(t)\sin(-\omega t)}_{\text{Imaginary part}}\, dt$$





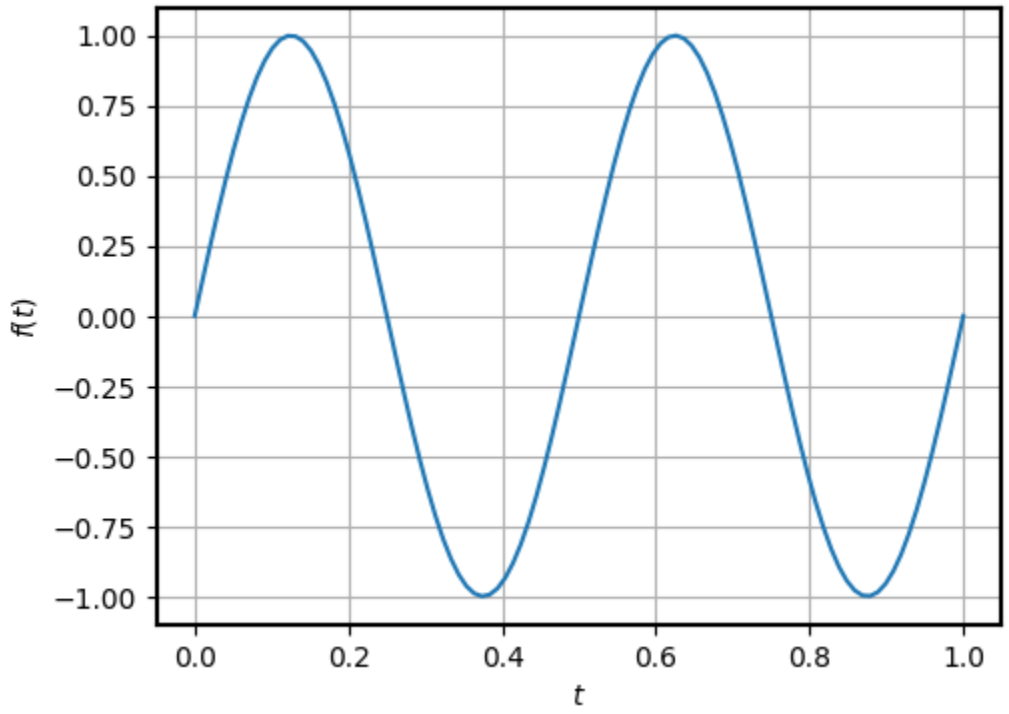
Magnitude & Phase



Example: A 2Hz Sine Wave

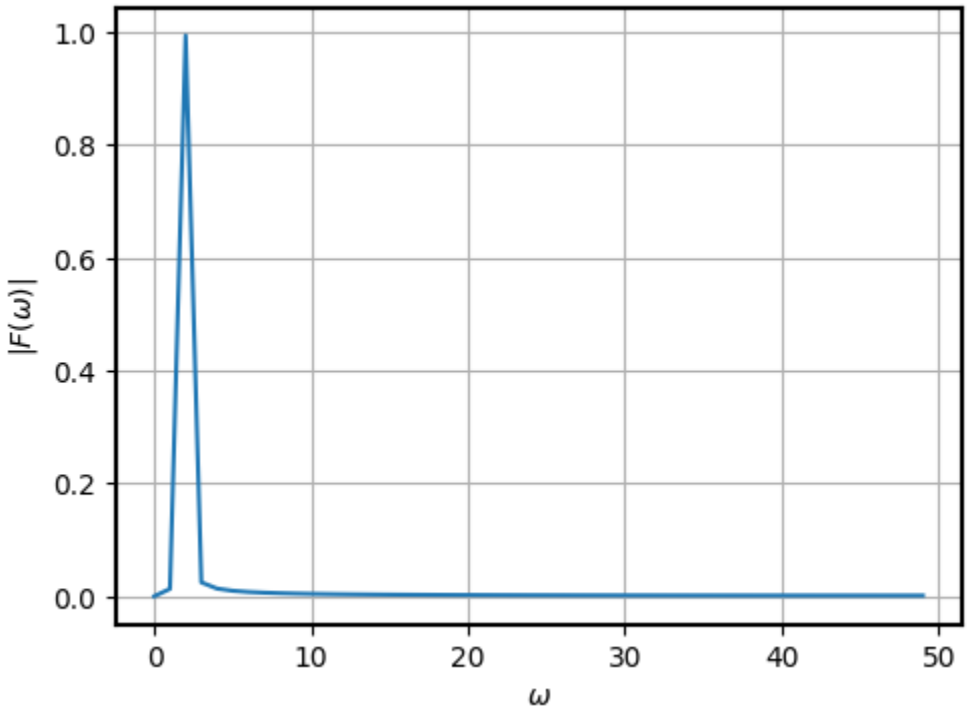
Signal

(time-domain)



Spectrum

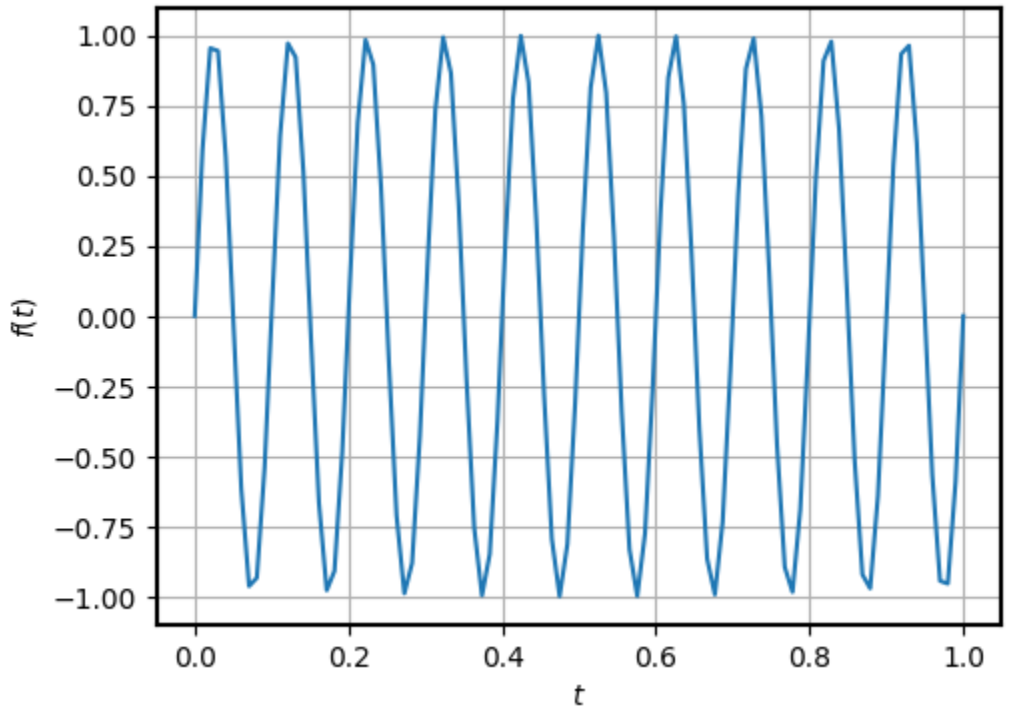
(frequency-domain)



Example : A 10Hz Sine Wave

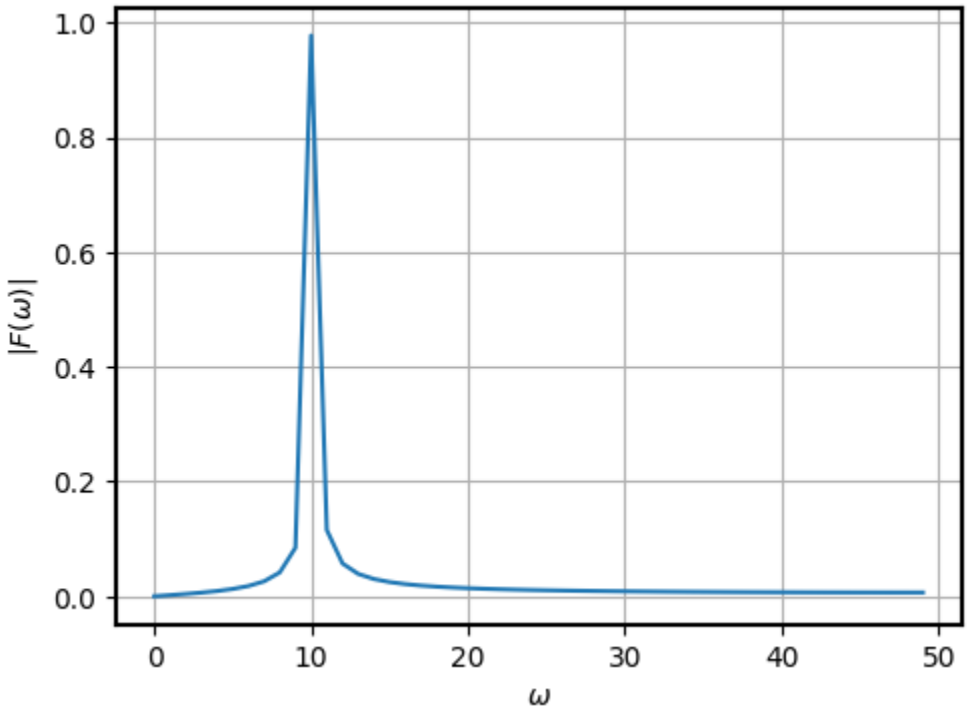
Signal

(time-domain)



Spectrum

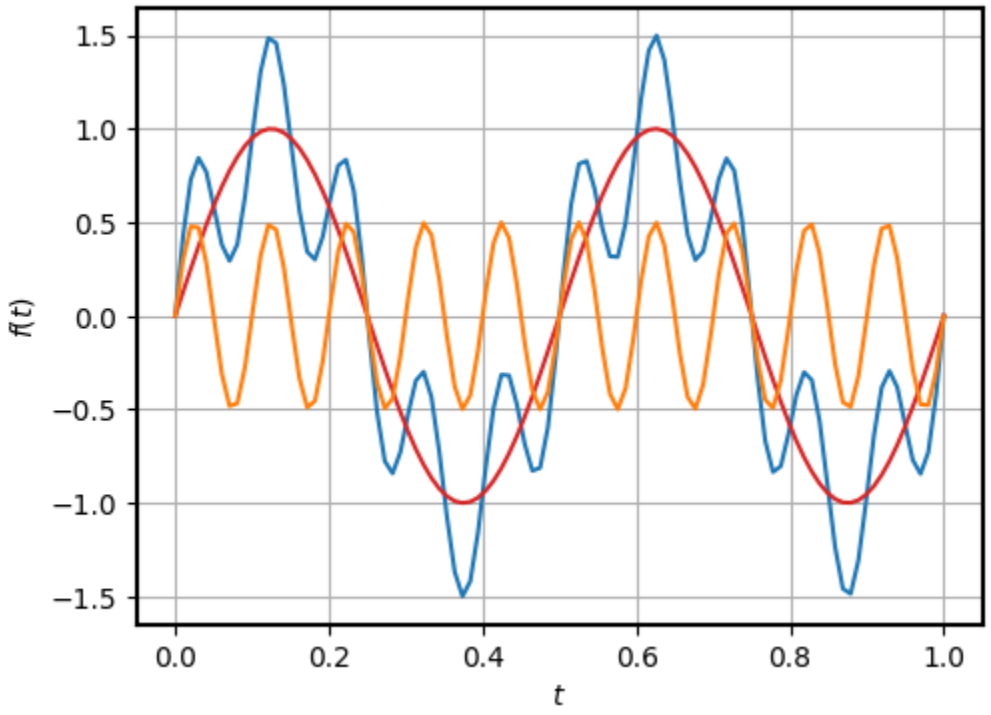
(frequency-domain)



Example: Sum of a 10Hz & 2Hz Sine Waves

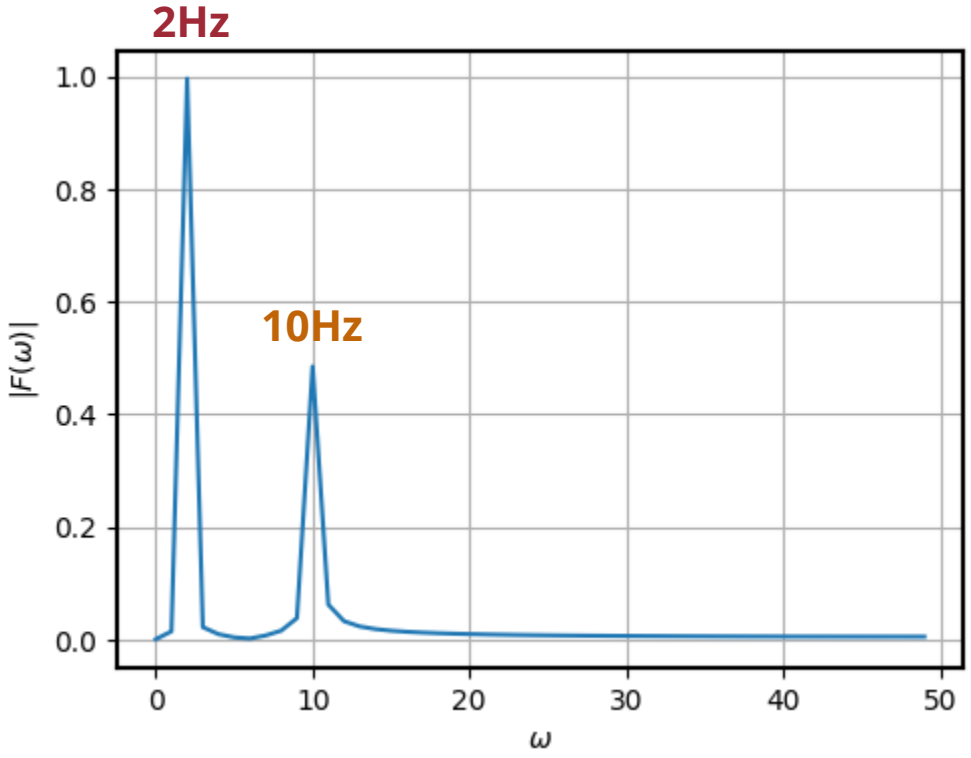
Signal

(time-domain)



Spectrum

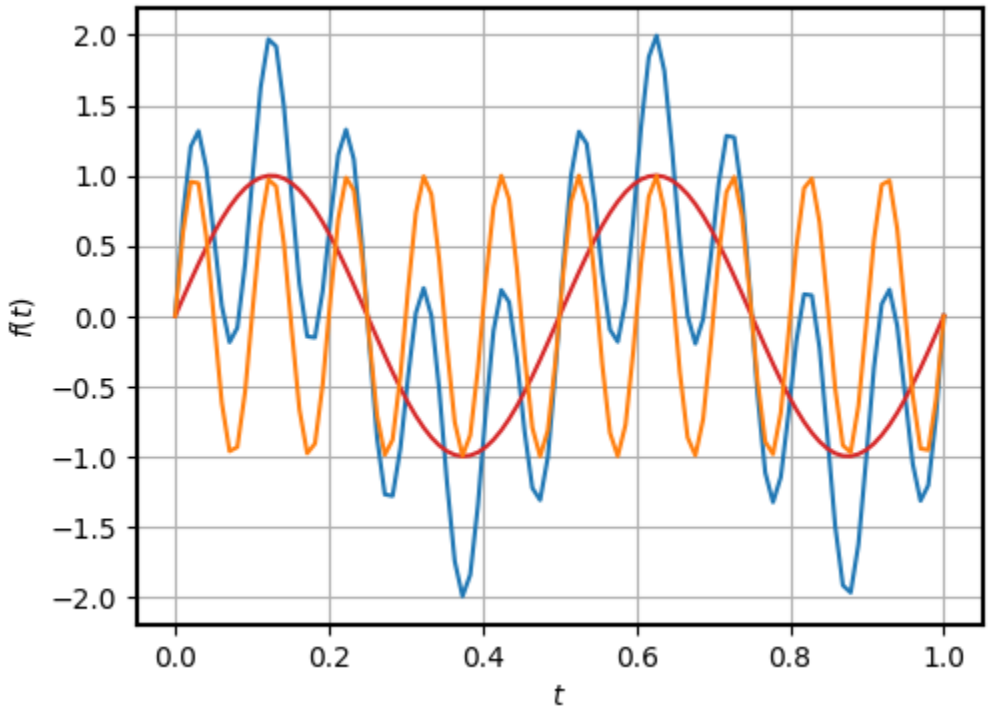
(frequency-domain)



How about this?

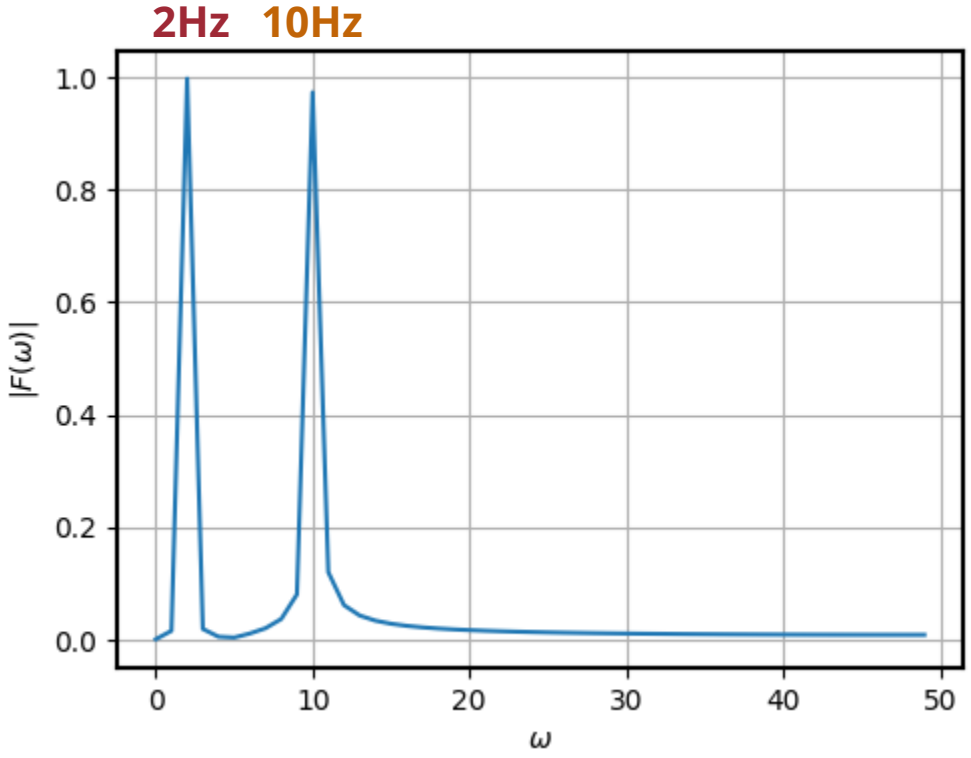
Signal

(time-domain)



Spectrum

(frequency-domain)



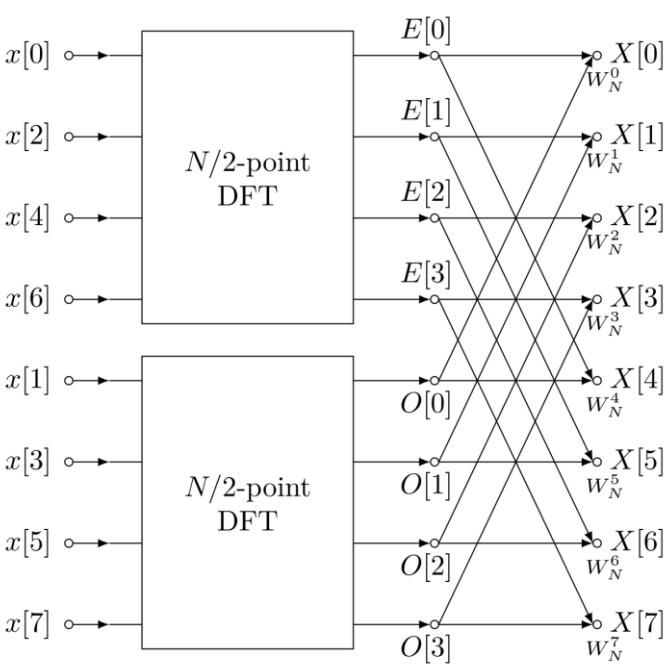
Discrete Fourier Transform (DFT)

- **Intuition**: Fourier transform with **discrete time and frequency**
  - Used for **digital audio** → we cannot achieve an infinite sampling rate…
- Mathematical formulation:

$$X_k = \sum_{n=0}^{N-1} x_n \; e^{-j2\pi \frac{k}{N} n}$$

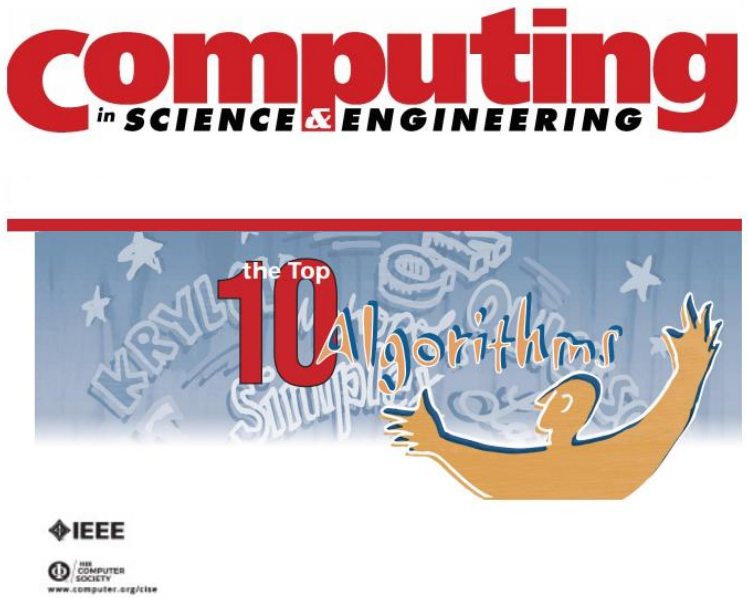
In Practice: Fast Fourier Transform (FFT)

- An efficient implementation of discrete Fourier transform
  - Reduce the complexity from $O(n^2)$ to $O(n \log n)$



(Source: Yangwenbo99 via Wikimedia)

Top 10 algorithms from the 20th century

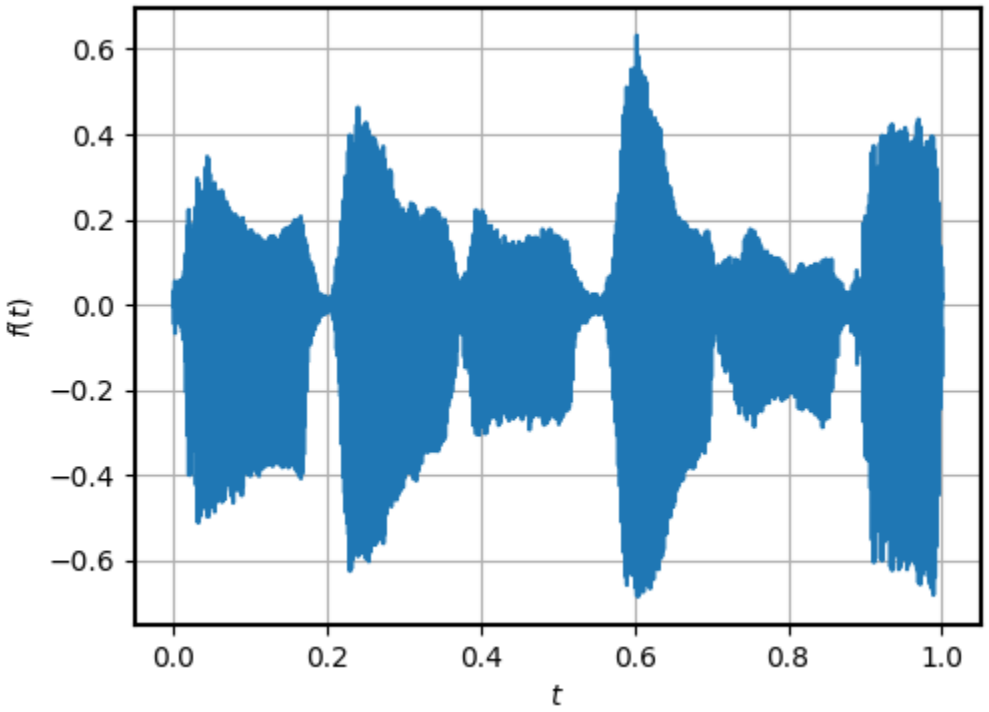


Time-Frequency Analysis

Fourier Transform of a Trumpet Sound

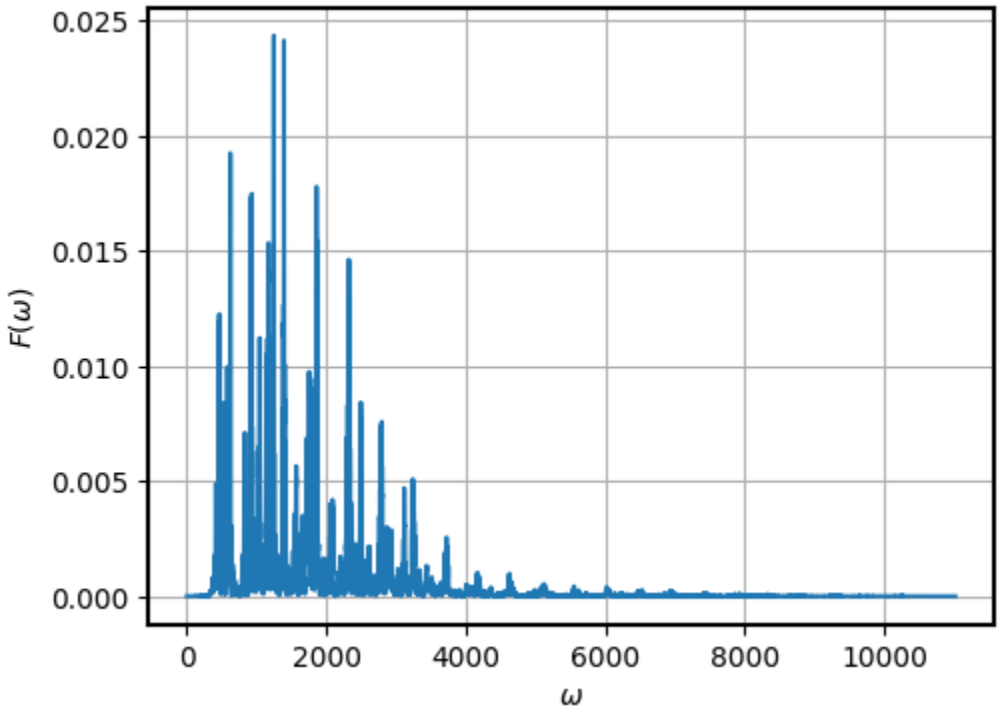
Signal

(time-domain)



Spectrum

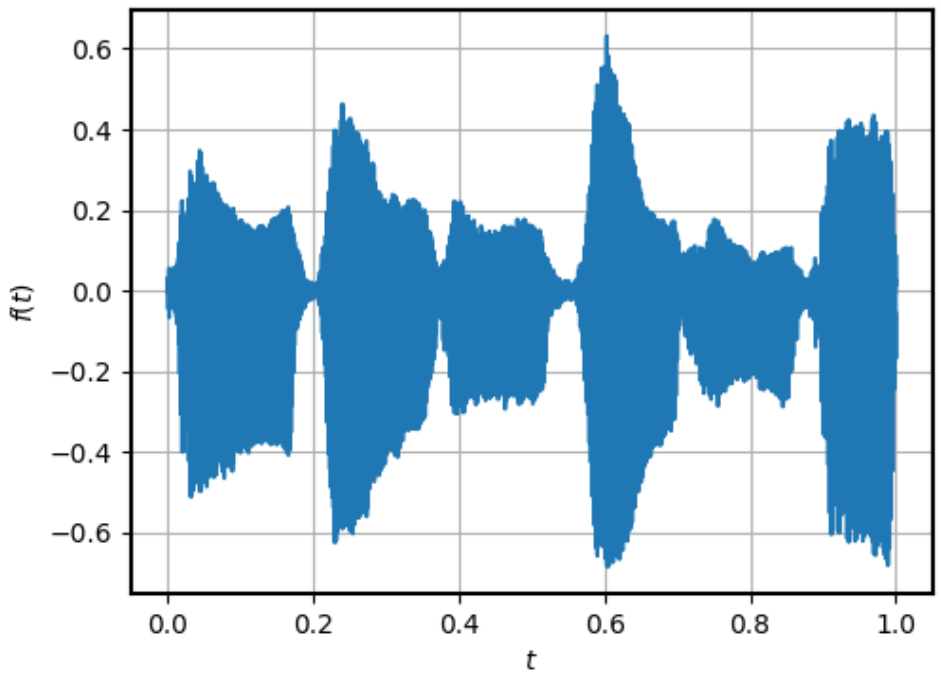
(frequency-domain)



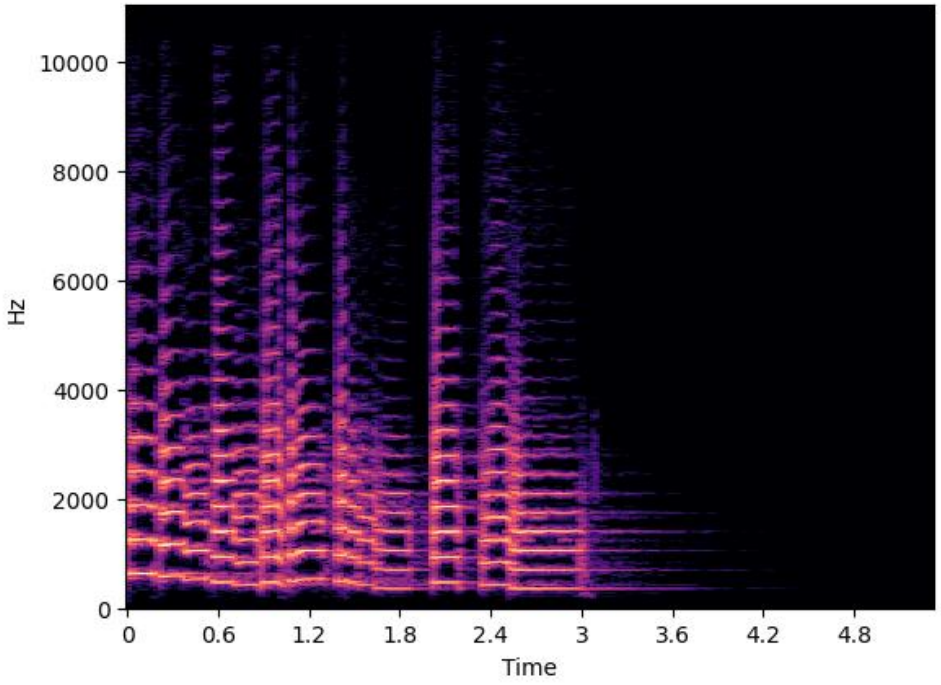
Fourier Transform cannot localize!

Short-Time Fourier Transform (STFT)

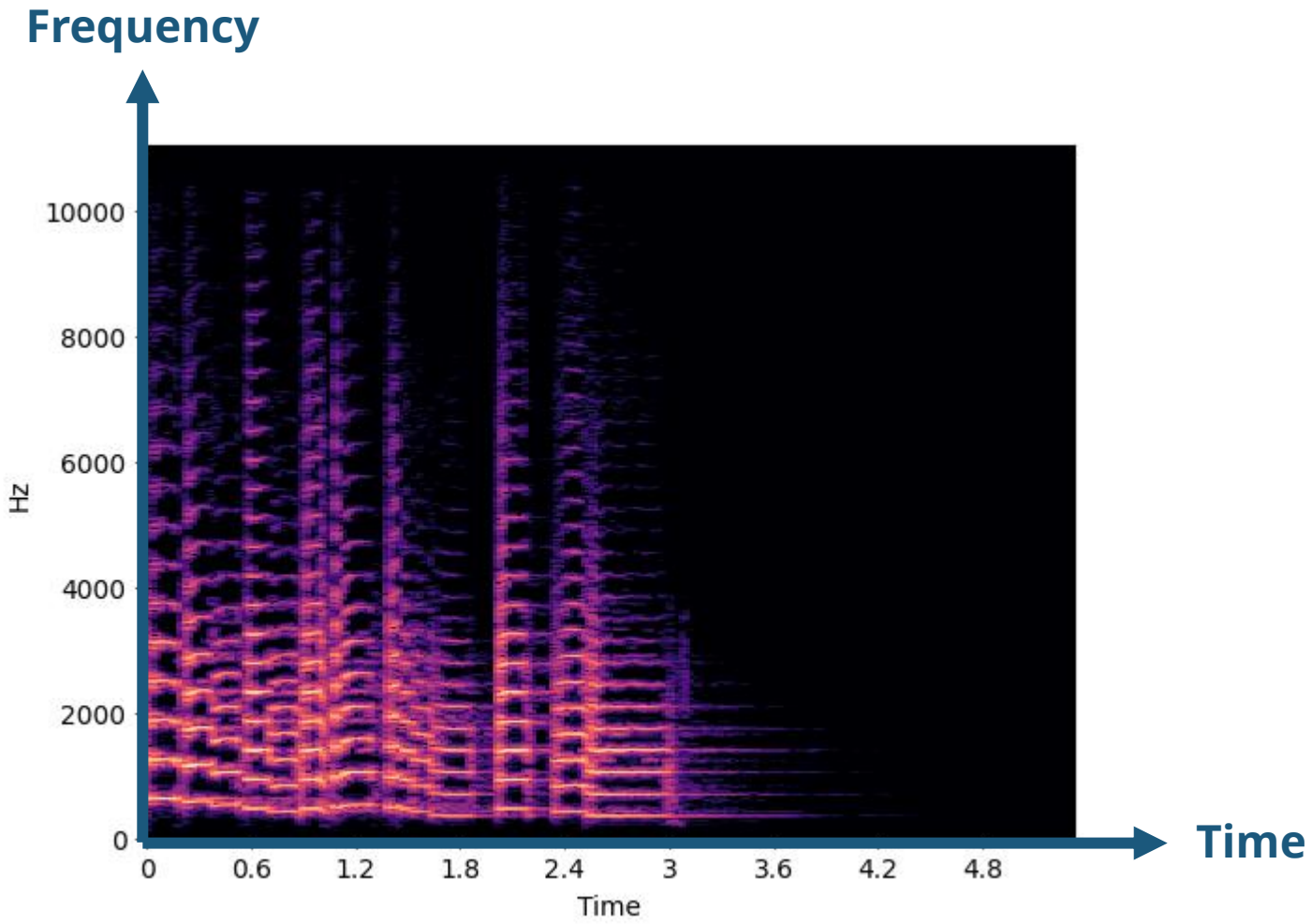
- **Intuition**: Slice the audio into chunks and apply Fourier transform



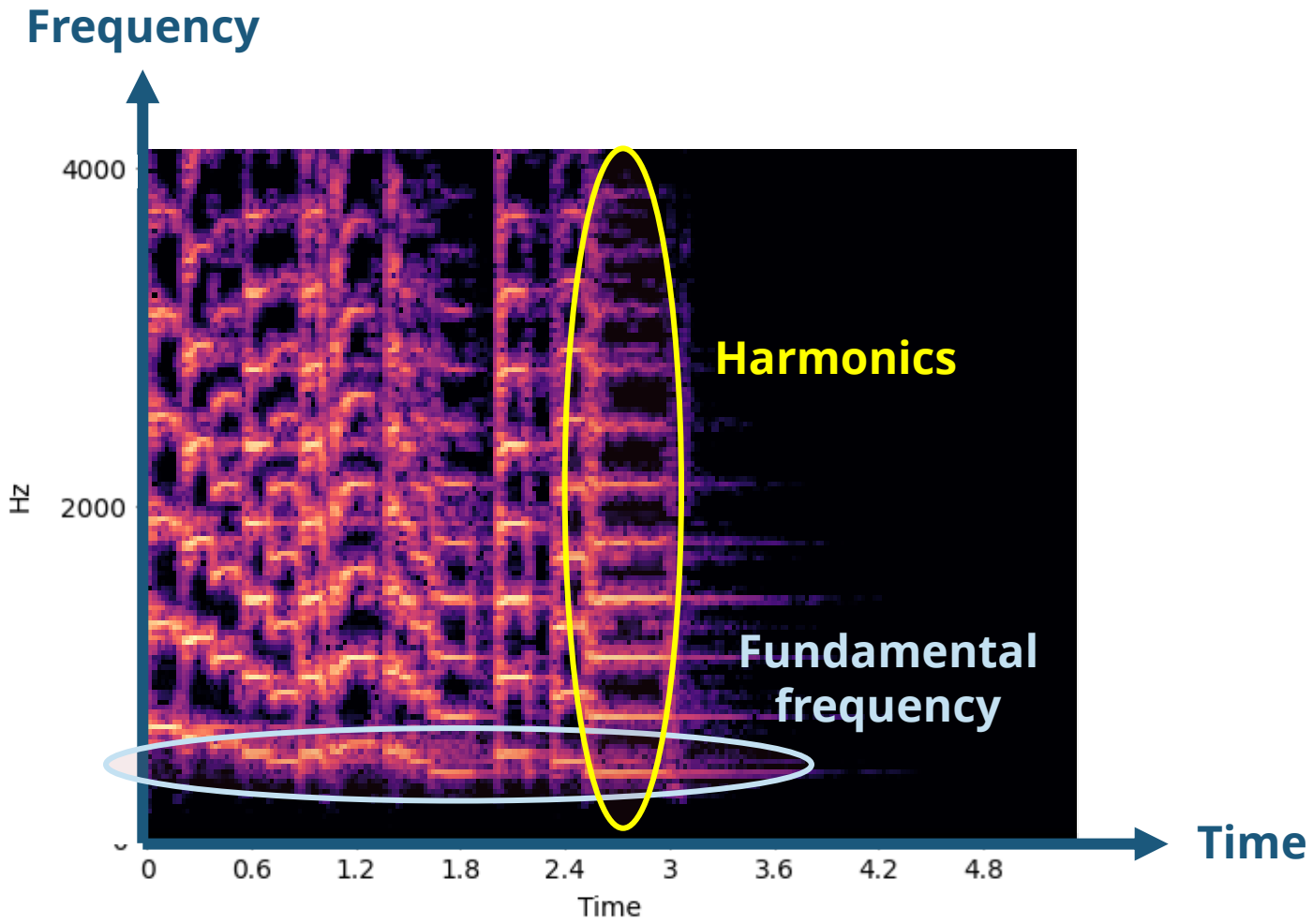
Short-Time Fourier Transform



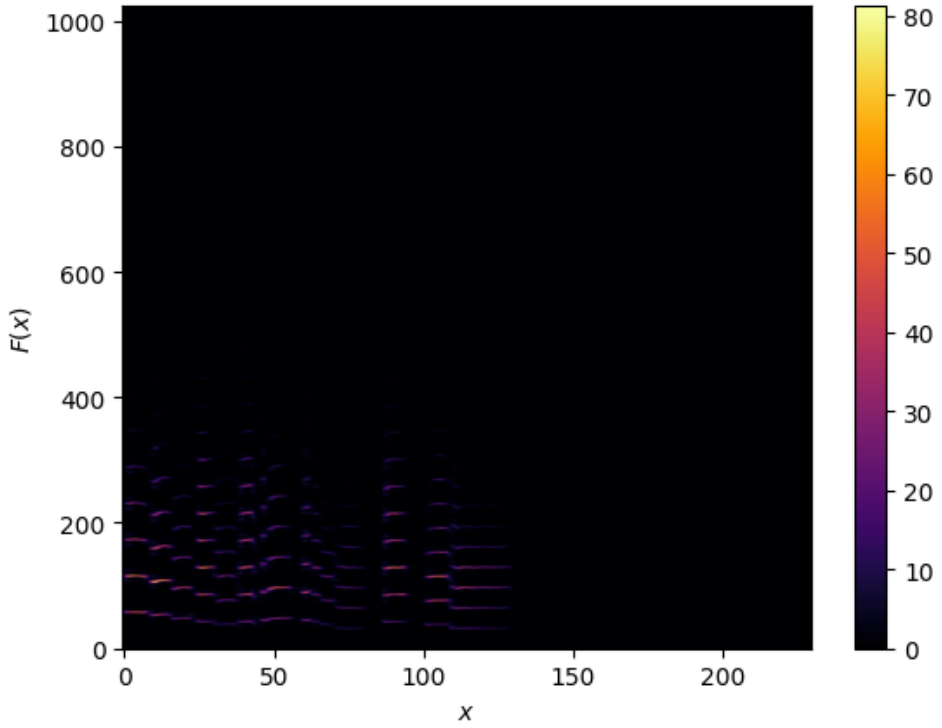
Spectrogram



Spectrogram



Example: `librosa.stft`

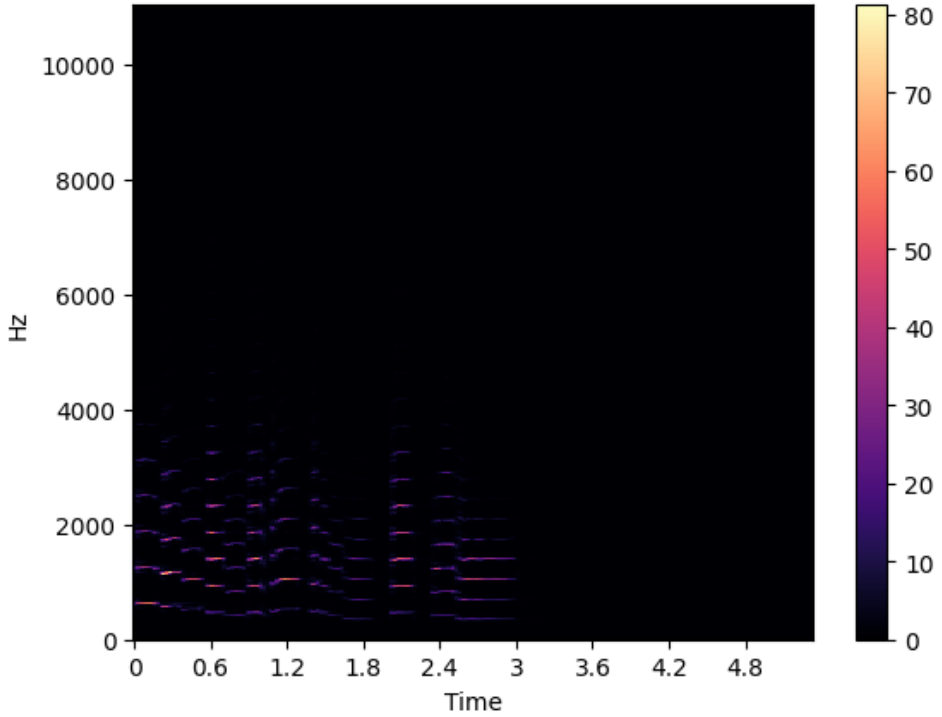



```
# Load the example audio in librosa
y, sr = librosa.load(librosa.example("trumpet"))

# Compute the spectrogram
S = np.abs(librosa.stft(y))

# Plot the spectrogram
im = plt.imshow(S, cmap="inferno", aspect="auto",
                origin="lower")
plt.colorbar(im)
plt.xlabel("Time (sec)")
plt.ylabel("Frequency (Hz))")
plt.show()
```

Example: librosa.display.specshow

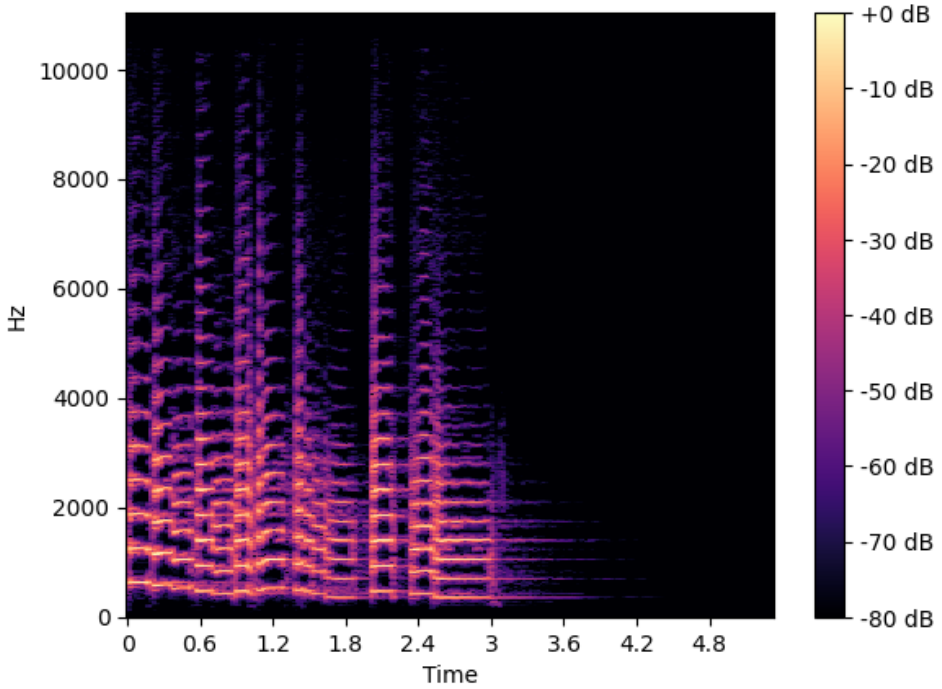



```
# Load the example audio in librosa
y, sr = librosa.load(librosa.example("trumpet"))

# Compute the spectrogram
S = np.abs(librosa.stft(y))

# Plot the spectrogram
im = librosa.display.specshow(S, x_axis="time",
                              y_axis="linear")
plt.colorbar(im)
plt.show()
```

Example: librosa.amplitude_to_db




```
# Load the example audio in librosa
y, sr = librosa.load(librosa.example("trumpet"))

# Compute the spectrogram
S = np.abs(librosa.stft(y))
S_db = librosa.amplitude_to_db(S, ref=np.max)

# Plot the spectrogram
im = librosa.display.specshow(S_db, x_axis="time",
                              y_axis="linear")
plt.colorbar(im, format="%+2.0f dB")
plt.show()
```

Timbre

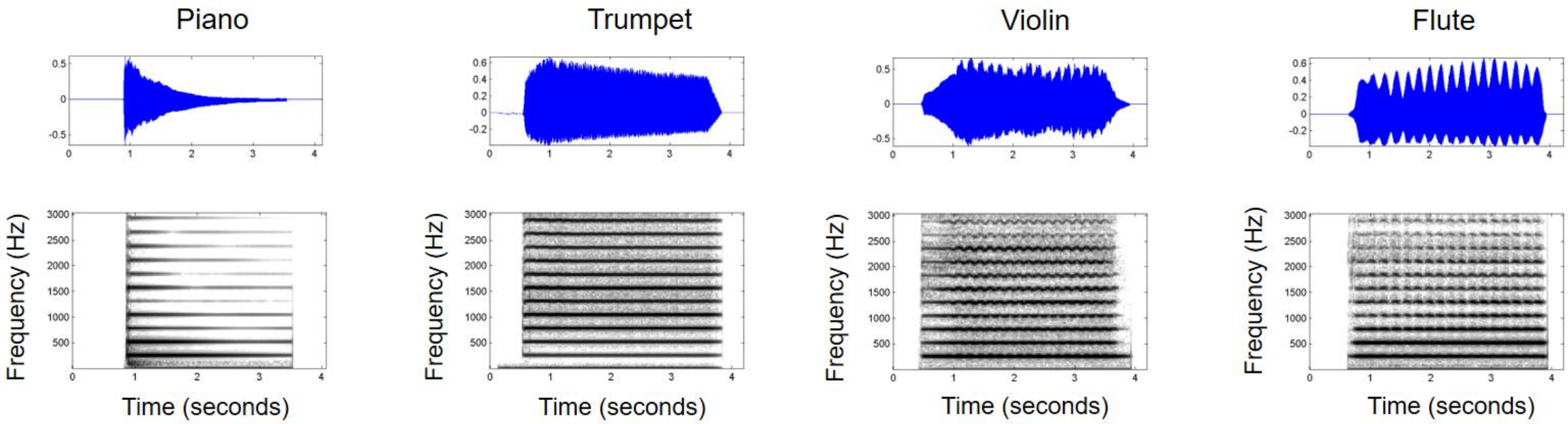
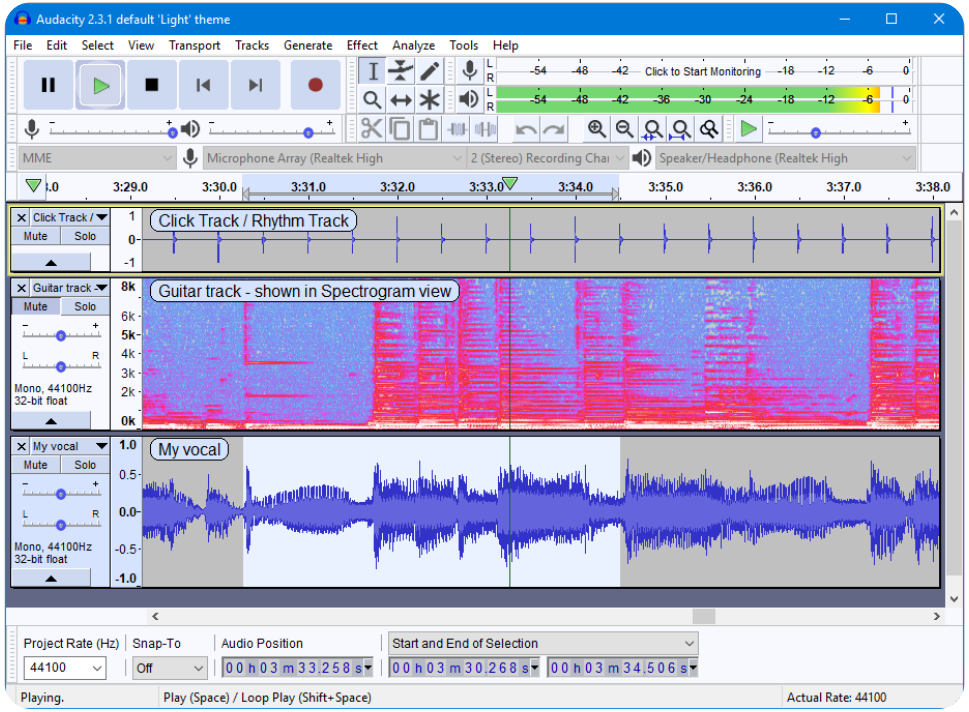


Figure 1.23 from [Müller, FMP, Springer 2015]

(Source: Müller et al., 2021)

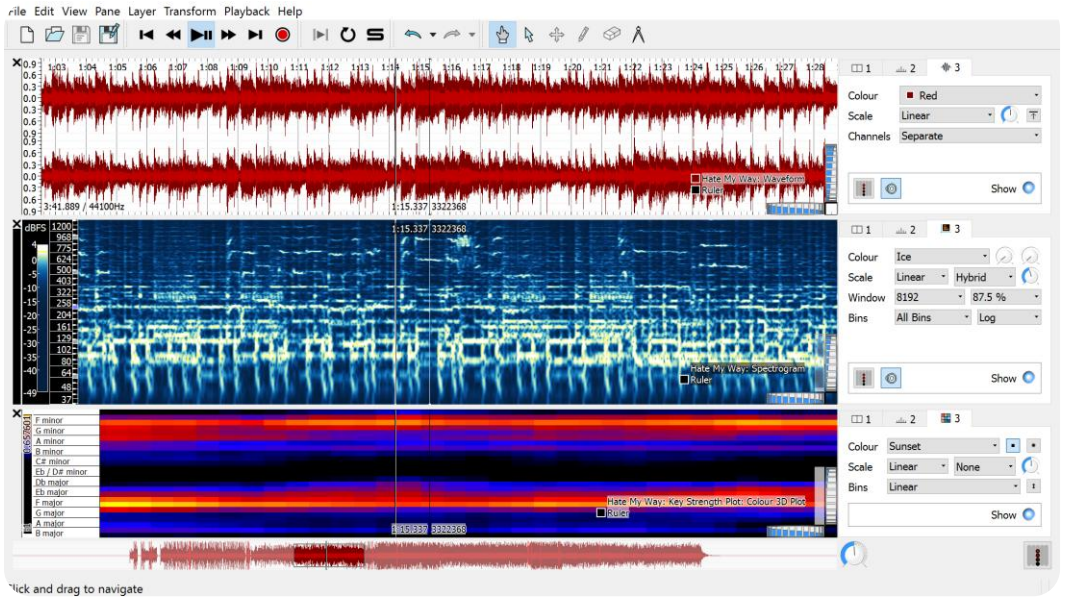
Software

Audacity



(Source: audacity-2.3.1 via Internet Archive)

Sonic Visualiser



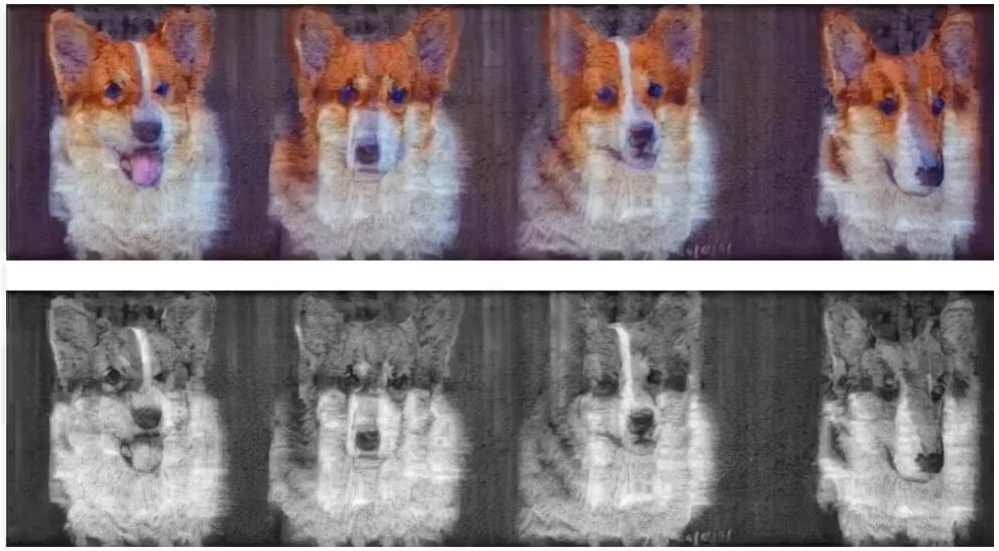
(Source: sonicvisualizer.org)

archive.org/details/audacity-2.3.1
sonicvisualiser.org

Images that Sound (Chen et al., 2024)

Using diffusion models to generate visual spectrograms that look like images but can also be played as sound.

Image prompt: a colorful photo of corgis



Audio prompt: dog barking

(Source: Chen et al., 2024)

Image prompt: a colorful photo of tigers

Audio prompt: tiger growling

(Source: Chen et al., 2024)

Ziyang Chen, Daniel Geng, and Andrew Owens, "Images that Sound: Composing Images and Sounds on a Single Canvas," *NeurIPS*, 2024.

Images that Sound (Chen et al., 2024)

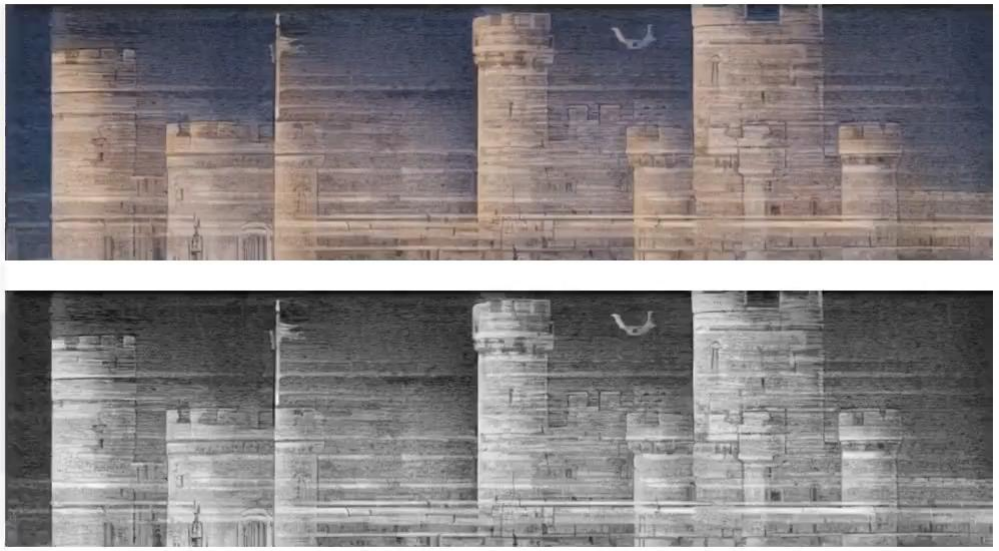
Using diffusion models to generate visual spectrograms that look like images but can also be played as sound.

Image prompt: a colorful photo of an auto racing game

Audio prompt: a race car passing by and disappearing

(Source: Chen et al., 2024)

Image prompt: a colorful photo of a castle with bell towers



Audio prompt: bell ringing

(Source: Chen et al., 2024)

Ziyang Chen, Daniel Geng, and Andrew Owens, "Images that Sound: Composing Images and Sounds on a Single Canvas," *NeurIPS*, 2024.