# Adapting Transformers to Audio Data

Transformers first reshaped NLP and then computer vision. The same principles now extend to **audio**, where sound is treated as a sequence that can be tokenized for self-attention. The central question is not whether transformers can work with audio, but **how the audio signal is represented before entering the model**.

---

## From Images to Audio Tokens

In vision, the breakthrough of Vision Transformers (ViTs) was to split images into **patches**, treat each as a token, and let self-attention model their relationships. Audio follows the same logic: it unfolds over time and can be divided into smaller units:

- **Frames** of a waveform (raw samples grouped into chunks), or
- **Spectrogram patches** (slices of a time–frequency map).

These units become tokens, allowing transformers to capture both local patterns and long-range temporal dependencies.

---

## Why Self-Attention Fits Audio

Earlier approaches like RNNs and LSTMs modeled sequences step by step. They were effective but struggled with:

- **Long-range dependencies** (e.g., linking intonation at the start of a sentence to emphasis at the end).
- **Scalability**, since sequential processing is slow.

Transformers overcome both limits. Self-attention lets the model link any two points in an audio sequence regardless of distance, while parallelization enables efficient training even on long recordings.

---

## Two Main Representation Paths

### 1. Raw Waveform Models

- Process audio directly from the signal.
- A small convolutional extractor converts waveforms into latent frames, then passed into a transformer.
- Example: **Wav2Vec 2.0**.
- Strengths: robust to noise, multilingual adaptability, avoids hand-engineered features.
- Applications: speech recognition, consumer voice analytics.

**2. Spectrogram-Based Models**

- Convert audio into a **time–frequency spectrogram**.
- Treated like an image: patches are embedded and fed into a transformer.
- Example: **Audio Spectrogram Transformer (AST)**.
- Strengths: interpretable (patterns visible in spectrograms), benefits from ViT transfer.
- Applications: sound event detection, music and emotion classification.

---

## Training Innovations

- **Masked prediction objectives** (e.g., **HuBERT**, **WavLM**) → predict missing audio segments, inspired by BERT.
- **Contrastive pretraining** (e.g., **CLAP**) → align audio with text in a joint embedding space for retrieval and multimodal tasks.

---

## Practical Hugging Face Examples

```
from transformers import pipeline

# Speech recognition (waveform → text)
asr = pipeline("automatic-speech-recognition",
model="facebook/wav2vec2-base-960h")
print(asr("sample.wav"))

# Audio classification (spectrogram → class labels)
clf = pipeline("audio-classification", model="MIT/ast-finetuned-
audioset-10-10-0.4593")
print(clf("sample.wav"))
```

---

## Takeaway

The transformer backbone remains unchanged; what differs is the **tokenization strategy**. Whether using raw waveform frames or spectrogram patches, self-attention learns the temporal and frequency patterns of sound. This makes transformers especially powerful for tasks in **psychology, sociology, cultural studies, and brand management**, where meaning depends not only on *what* is said, but also *how* it sounds across time.

---