Google

From Paper to Production

The Engineering and Standardization Challenges of Neural Speech Coding

Jean-Marc Valin

May 4, 2026

Introduction

Neural codecs provide undeniable benefits over traditional speech codecs

They have so far been confined to the lab (paper codecs) or custom apps

How do we get to mass deployment of a standard neural codec?

Google

Outline

A brief history

The problems we're facing

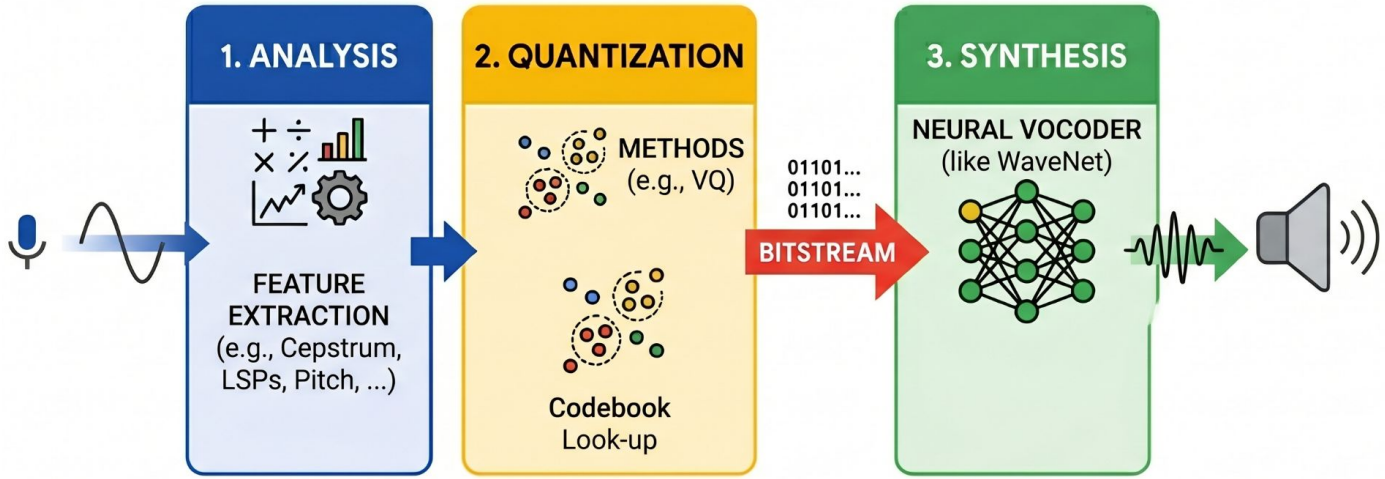
What we need

Where to go from there

Google

First Gen Neural Speech Codecs

- Started in 2018 (Kleijn et al.)
  - Classical features+quantization, neural vocoder



Google

First Gen Neural Speech Codecs

- Pros
  - Better than all previous low-bitrate codecs
  - Simple bitstream
  - Easy “upgrade” through vocoder
- Cons
  - Complexity originally hundreds of GFLOPS
  - Limited intelligibility gain (decoupled from quality)
  - Poor robustness to noise/reverb

Google

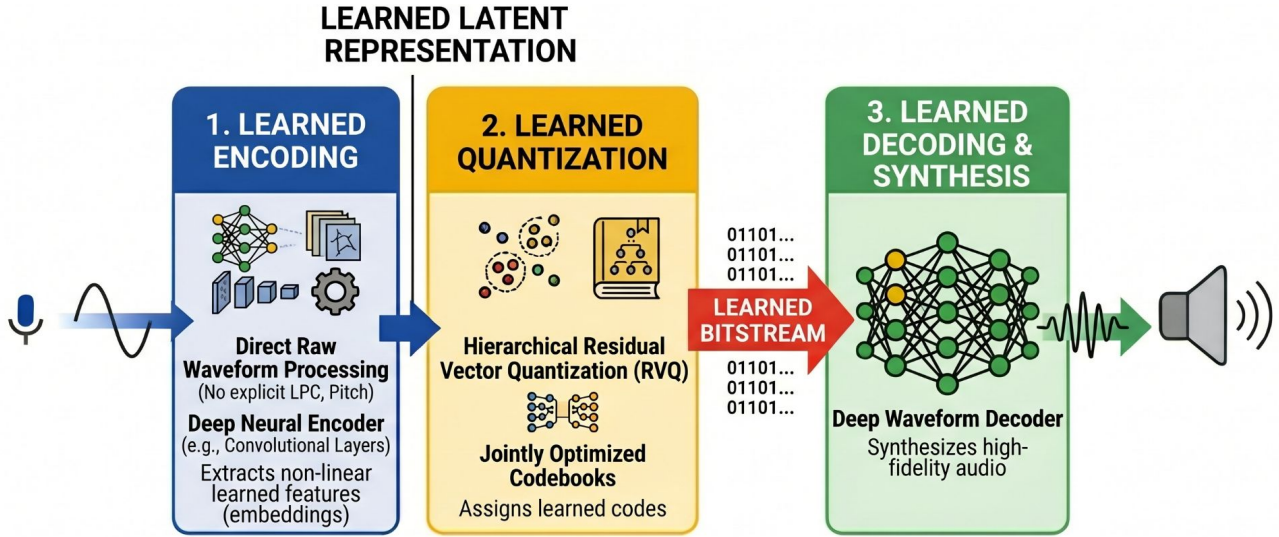
Vocoder Upgrades

- First gen: WaveNet/SampleRNN
  - 100+ GFLOPS
- Neural vocoders have improved
  - WaveRNN (2018): 10 GFLOPS
  - LPCNet (2019): 3 GFLOPS
  - FARGAN (2024): 0.6 GFLOPS
- Can keep improving without changing the feature definition

Google

End-to-End Speech Codecs

- A new class of neural codecs, e.g. SoundStream
  - Features and quantization can be learned



Google

End-to-End Speech Codecs

- Pros
  - Even better quality than “classical” neural codecs
    - Can scale to transparency
  - Features and quantization can be learned
- But nothing’s perfect
  - Often very large models
  - Robustness issues with out-of-domain audio
  - Not interpretable (big blob of weights)

Google

Why Have Standards?

Interoperability

They allow interoperability between a wide range of devices

Predictability

Their behaviour is well understood and characterized

Longevity

Standards stay around for a long time

33 years
JPEG

54 years
G.711

Google

It has been 8 years…

Where are the neural codec standards?

Google

The Paper Codec

"Here's a 100 MB blob, will you make it a standard?"

- "It's great and I'll have an even better architecture next year"
- "It cannot reach good quality, but at 1 kb/s it's the best codec out there"

Google

What We Need

Low Resource
Complexity and memory efficiency for broad device support

Flexible Operating Points
Adaptability to varying bitrates and performance needs

Interpretability
Clear understanding of model behavior and decisions

Robustness
Reliable performance under diverse and challenging conditions

Interoperable Evolution
Forward compatibility throughout the system's lifetime

Compelling Use Case
Demonstrable value and real-world application

Google

Low Resource

Device Compatibility

Low enough complexity to run easily on old/cheap phones
Slow cores, slow memory

Small Footprint

Optimized app download size

Google

Flexible Operating Points

Dynamic Bitrates

Wide range of switchable bitrates, from lowest achievable to near transparency

Universal

Switchable bandwidth, frame size, and complexity
Can it (almost) replace Opus?

Not going back to the old days of specialized codecs

Google

Interpretability

Open question: how can we make sense of a neural codec?

- Meaningful architecture?
- Differentiable DSP?
- Training constraints?

Interpretability will help:

Robustness

Long term evolution

Resource efficiency

Google

Robustness

Operational Needs

Needs to just work in any situation:

- Noise, acoustics, and microphones
- Diverse languages and accents
- Resilience to packet loss

Key Factors

Robustness is primarily influenced by:

- Training data diversity
- System architecture

Google

Interoperable Evolution

Long-term Standards

Meant to be used over a long period of time

- Requires stable, mature technology
- You don’t get a do-over next year

Evolutionary Space

The best standards leave room to evolve while maintaining compatibility

- Example: original vs latest MP3 encoders

Google

Compelling Use Case

Header Overhead

RTC headers typically 40+ bytes

- 16 kb/s of headers using 20-ms frames
- 2 kb/s speech is useless for video calls

Standard Gap

New solutions need clear benefits over existing standards to gain traction

Examples

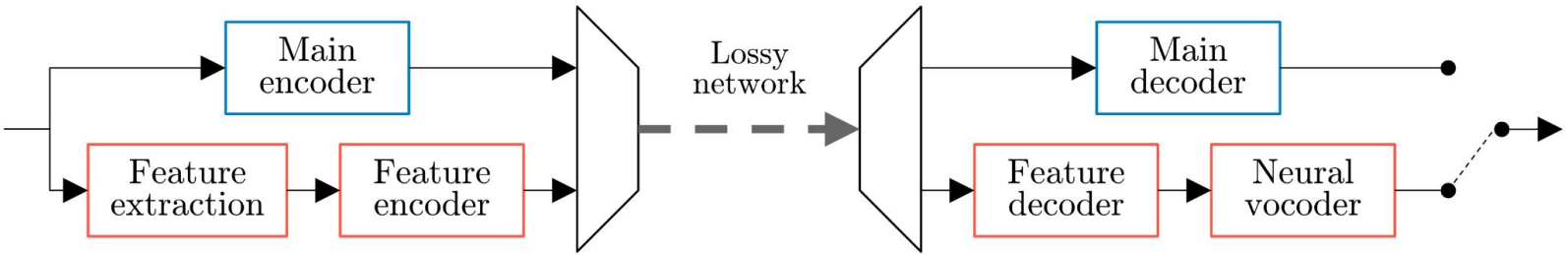
Existing use cases for niche pseudo-codecs:

- **DRED:** Massive redundancy in Opus
- **RADE:** "Analog codec" for HF Ham radio

Google

(Almost) Case Study: Deep REDundancy

- Designed to code only redundancy at ultra low-bitrate
  - Hybrid classical/end-to-end architecture
- Achieves most of the goals above because not a full codec
  - Under standardization at IETF (Opus extension)
  - First step to full neural codec



Google

What Can We Do? (Call to Action)

- Organize workshop on low-resource audio codecs
- Academic focus on practical aspects (scalability, flexibility, robustness)
- More research into interpretable/differentiable DSP
- Some public trial and error

Google

Conclusion

Advanced Tech

We already have great neural codec technology at our disposal

Standard Gap

We don’t yet have a standards ready to deploy

Path Forward

Many small steps are needed to get there

Google

Any Questions?

Google