Speech Coding Enhancement for Opus

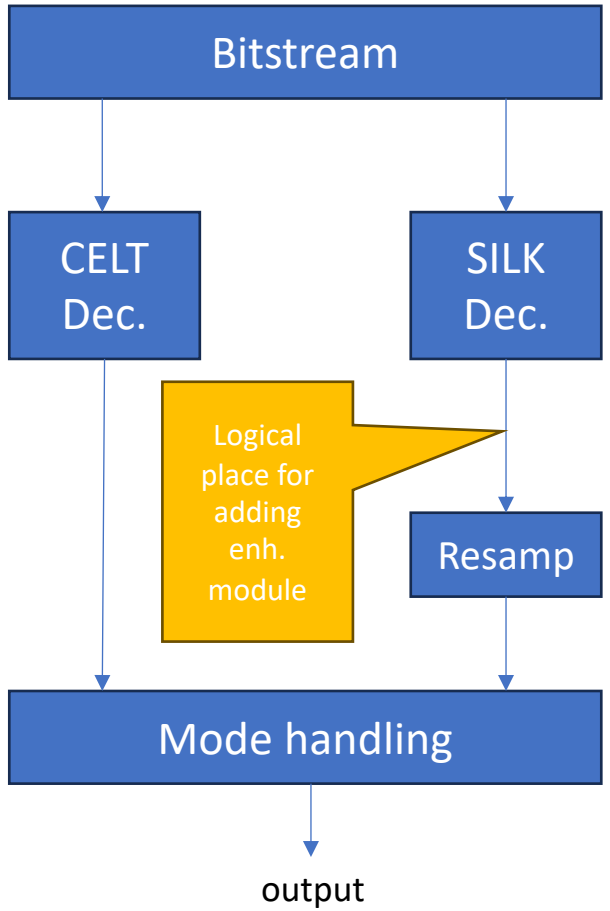
Presenter: Jan Buethe (AWS)

jbuethe@amazon.com

IETF 117

Enhancement in the Opus Decoder

- SILK: speech coding module
- CELT: MDCT based general audio coding
- Mode handling:
  - Switching between SILK and CELT
  - Combining low band from SILK decoder and high band from CELT decoder in hybrid mode
- Logical place for adding speech coding enhancement: between SILK decoder and resampling block



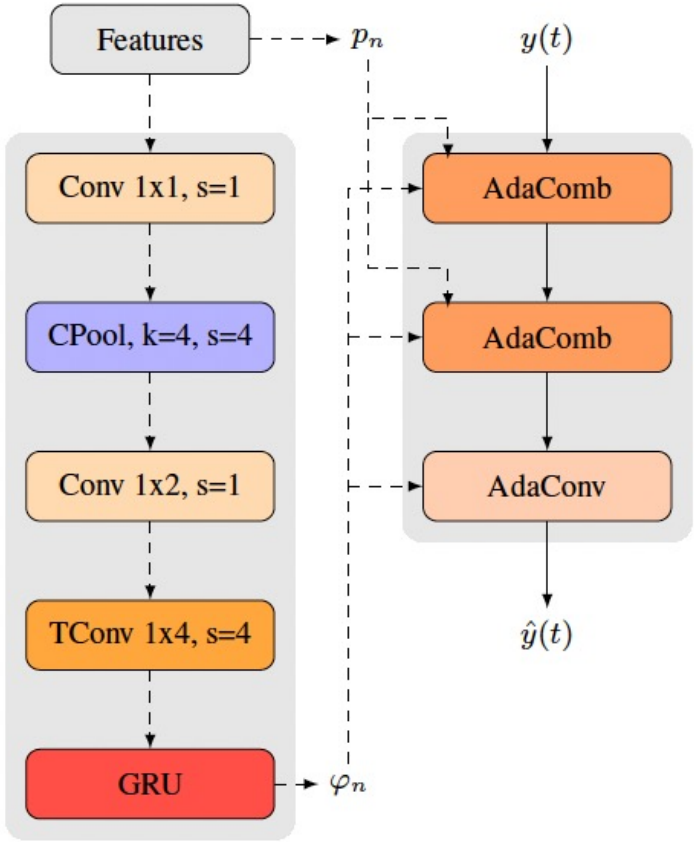
Challenges

1. Heterogeneous Input: Enhancement method must work for all content, all bitrates, and all encoder settings
2. Decoder Integrity: Addition of enhancement method must preserve decoder functionality (mode switching, hybrid mode)
3. Interoperability:
   - Encoder provides much freedom for tuning
   - Presence of enhancement module in decoder likely changes optimal encoding choices
   - Shifting to new optimum can break interoperability with legacy decoders

   Basic interoperability needs to be preserved!

Linear-Adaptive Coding Enhancer

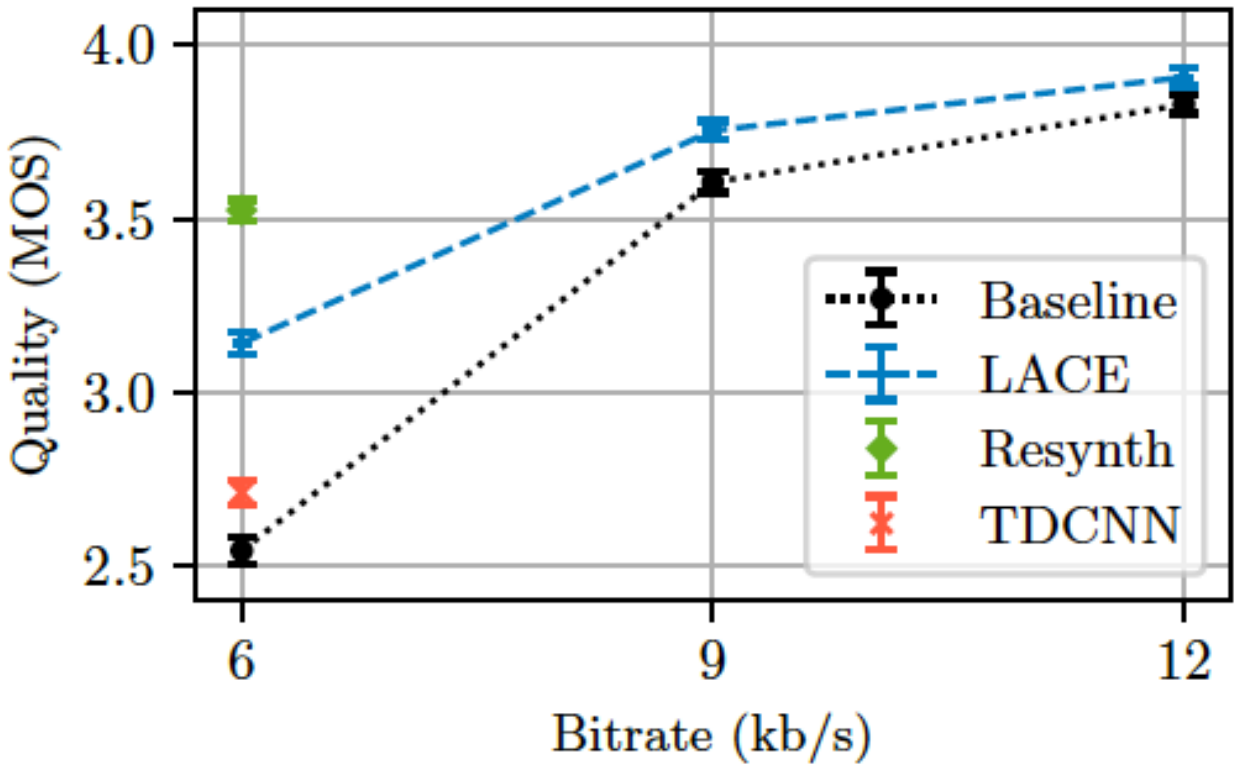
- LACE combines classic enhancement approaches (long-term / short-term filtering) with DNNs
- Linear but time varying filtering in signal path
- Filters computed from input features via DNN
- Complexity: 100 MFLOPS
- Model size: 300 K parameters
- Paper: https://arxiv.org/abs/2307.06610
- Demo: https://282fd5fa7.github.io/LACE/
- Python code: https://gitlab.xiph.org/xiph/opus/-/tree/opus-ng/dnn/torch/osce



How LACE addresses Challenges

1. Heterogeneous Input: Trained on multi-lingual dataset with random bitrate switching (6 to 64 kb/s) and random switching of encoding parameters
2. Decoder Integrity: Adds no delay and is trained to be approximately phase preserving -> can be seamlessly integrated
3. Interoperability: Wideband encoding extended to 6 kb/s. Gives low quality with legacy decoder but output is still intelligible

Evaluation (P.808)

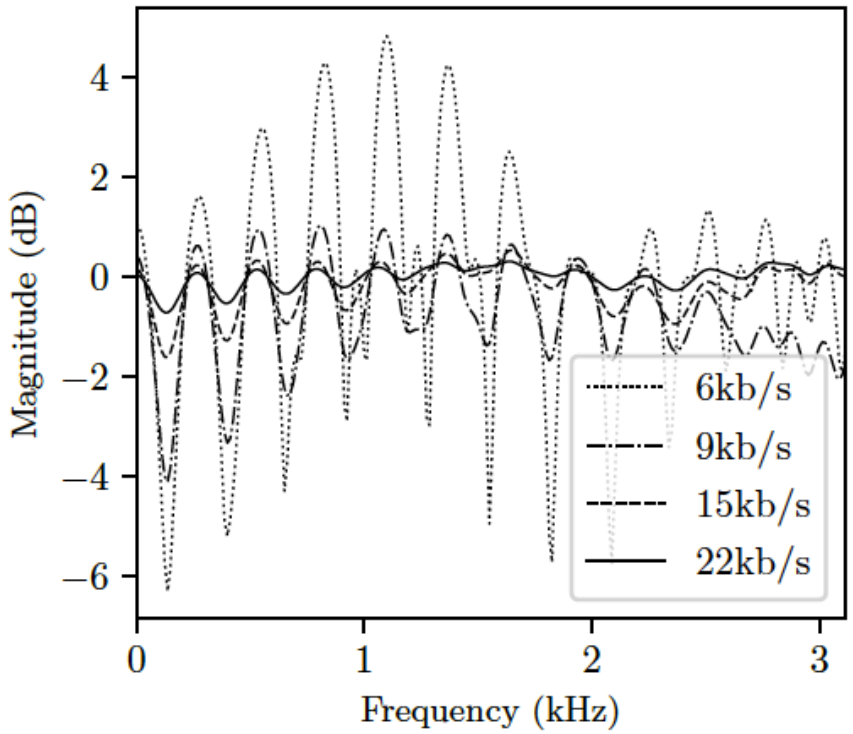


- Resynth (LPCNet resynthesis)
  - Complexity: ~3 GLOPS
  - Delay: 25 ms
  - Limited bitrate scalability
- LACE
  - Complexity: 100 MFLOPS
  - Delay: 0 ms
  - Full bitrate scalable

How to standardize?

- Any fixed model might be quickly outdated
- Application-dependent complexity / size /quality trade-off
- Proposal: specify requirements instead of (or on top of) method that ensure that
  1. enhancement model meets quality requirements,
  2. decoder functions well with enhancement model added, and
  3. basic interoperability is preserved for adapted encoder settings

Thank you!


Magnitude (dB)
4
2
0
−2
−4
−6
0
1
2
3
Frequency (kHz)
6kb/s
9kb/s
15kb/s
22kb/s