Deep Generative Models for Speech Compression

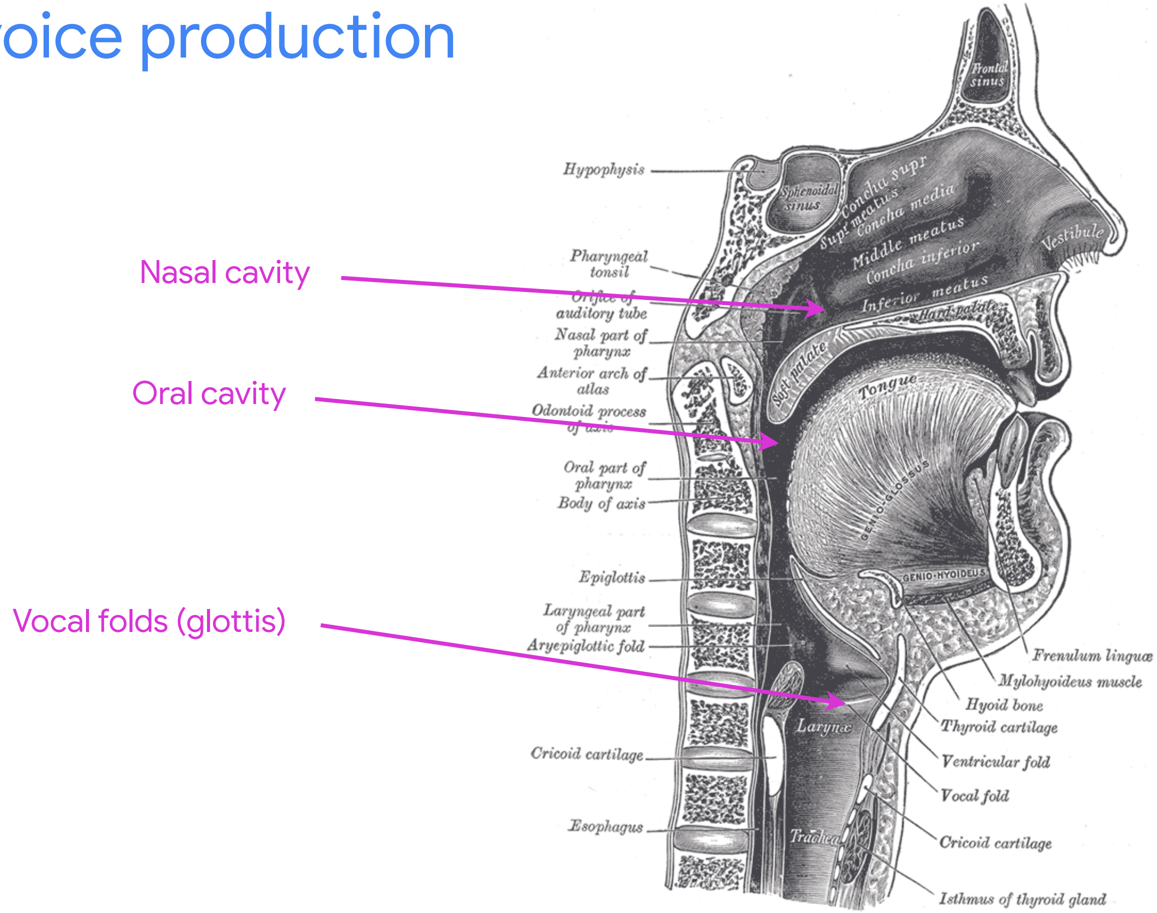
Jan Skoglund - Chrome Media Audio

Outline of today

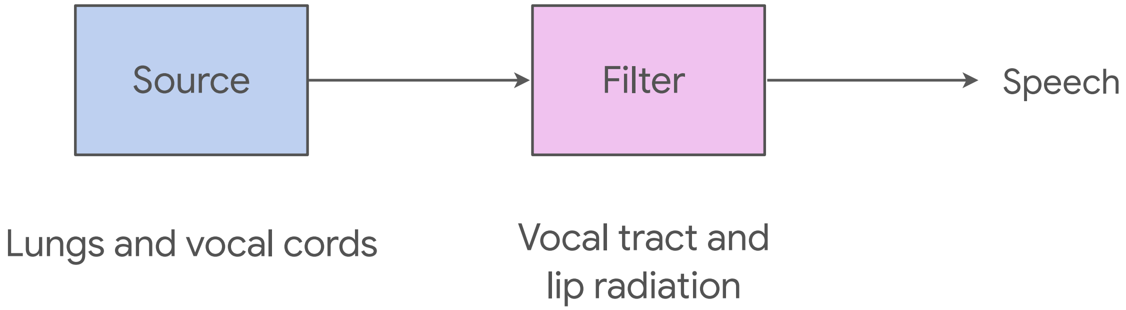
- Source-filter modeling for speech synthesis
- Speech coding
- Linear predictive coding
- Generative neural synthesis for coding
- LPCNet
- Noise-robust neural vocoding

- **Source-filter modeling for speech synthesis**
- Speech coding
- Linear predictive coding
- Generative neural synthesis for coding
- LPCNet
- Noise-robust neural vocoding

Human voice production



Source-filter modeling



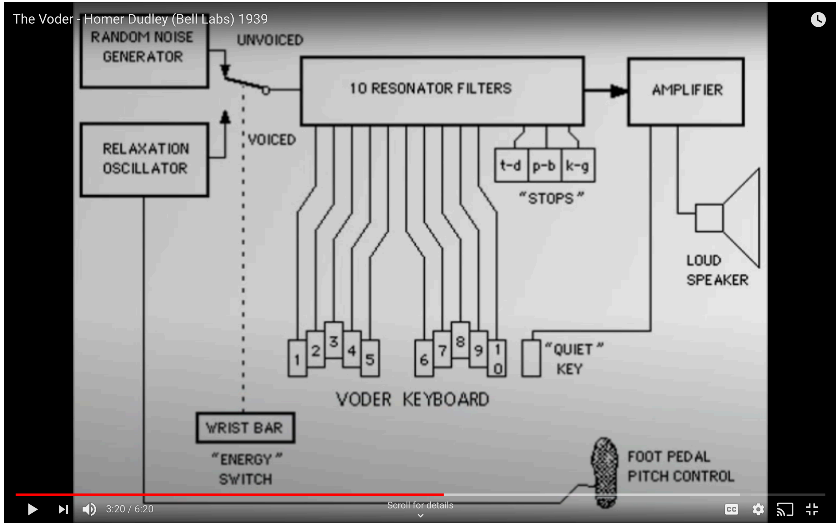
Mechanical synthesis

Von Kempelen's talking machine 1769-1804

- Mostly voiced synthesis
- Kitchen bellows - “glottis”
- Bagpipe reeds “glottis"
- Constant pitch
- Vowels formed by hands in front of rubber “mouth”
- [Video]

Electronic synthesis

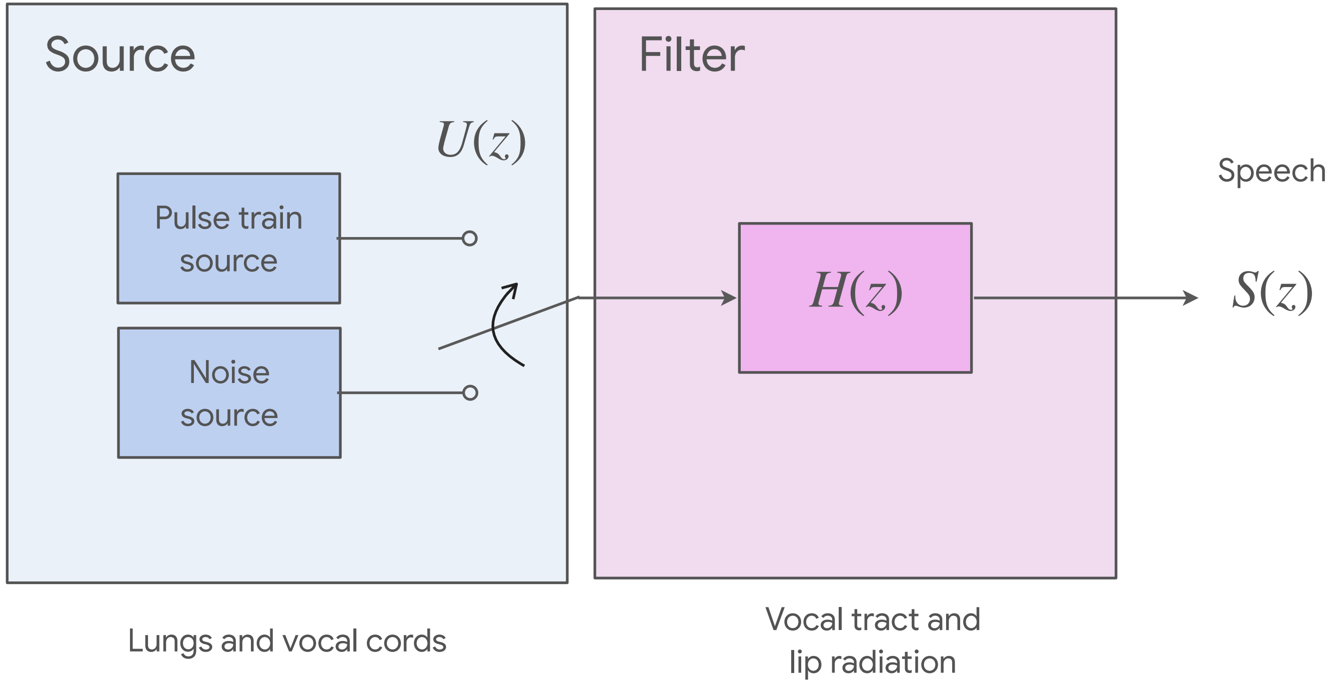
The Voder, Dudley 1939



- Full articulatory synthesis
- Required highly trained operators ~10 phonemes/s
- Part of Bell Labs "Vocoder" (channel vocoder) project
- Both analysis and synthesis
- Video

Source-filter modeling

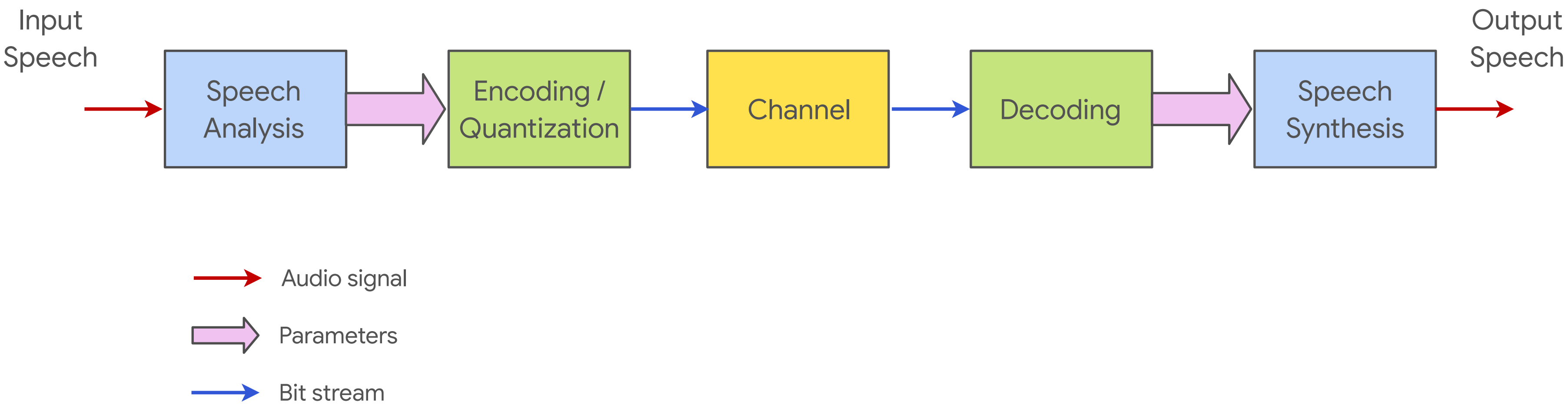
Signal processing view



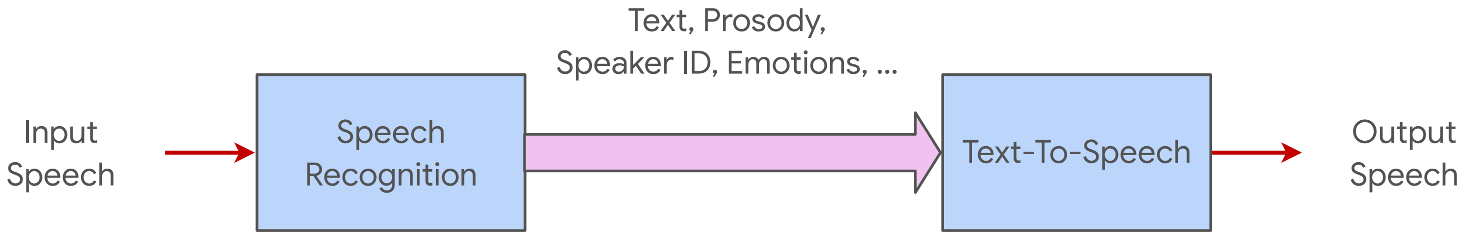
- Source-filter modeling for speech synthesis
- **Speech coding**
- Linear predictive coding
- Generative neural synthesis for coding
- LPCNet
- Noise-robust neural vocoding

Speech coding

Compression of speech for transmission and storage



Very low bit rate of speech - lower limit



- Early estimates (1950's)

  Shannon's lexical approach: ~50 bps

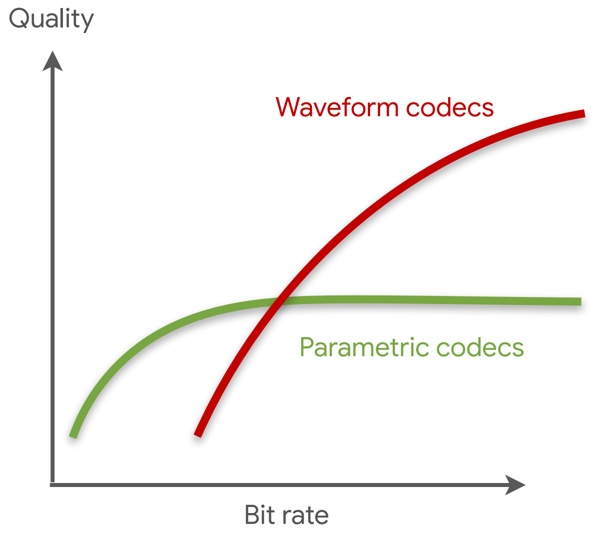
  Fano's noisy acoustical channel approach: ~1600 bps

- Recent estimate (2017) [1]: ~100 bps

Theoretical limit estimates assume very long delay

[1] *Van Kuyk, W. B. Kleijn, R. C. Hendricks*, "On the information rate of speech communication," ICASSP 2017

Practical speech coding



Parametric codecs, vocoders

- Bit rates from ~300 bps to ~5 kbps
- Quality limited to model
- Mostly narrowband speech (8 kHz sampling)

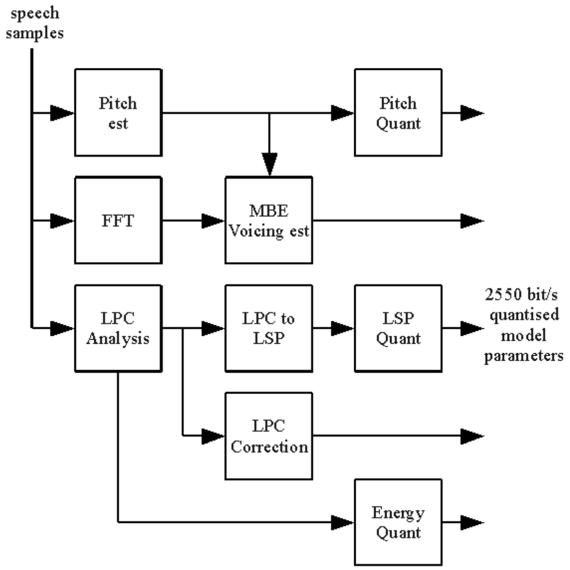
Waveform(-matching) coders

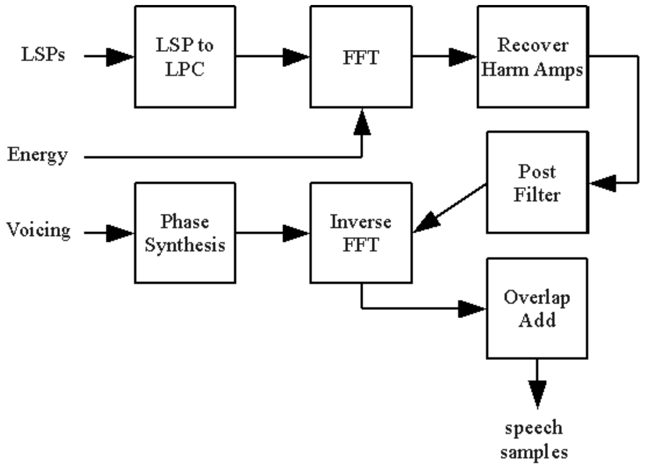
- Bit rates from ~3 kbps to ~100 kbps
- No limit in quality with increasing rate
- Narrowband, wideband (16 kHz sampling) and fullband (>32 kHz sampling)

Codec 2

Source-filter vocoding in the frequency domain

- Open-source codec by David Rowe [2]
- Bit rates from 450 bps to 3.2 kbps



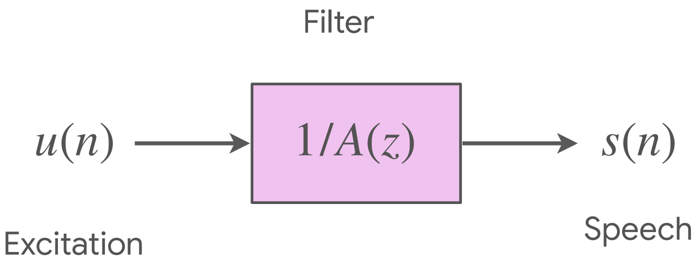


[2] https://github.com/drowe67/codec2

- Source-filter modeling for speech synthesis
- Speech coding
- **Linear predictive coding**
- Generative neural synthesis for coding
- LPCNet
- Noise-robust neural vocoding

Linear prediction analysis (LPC)

All-pole filter modeling



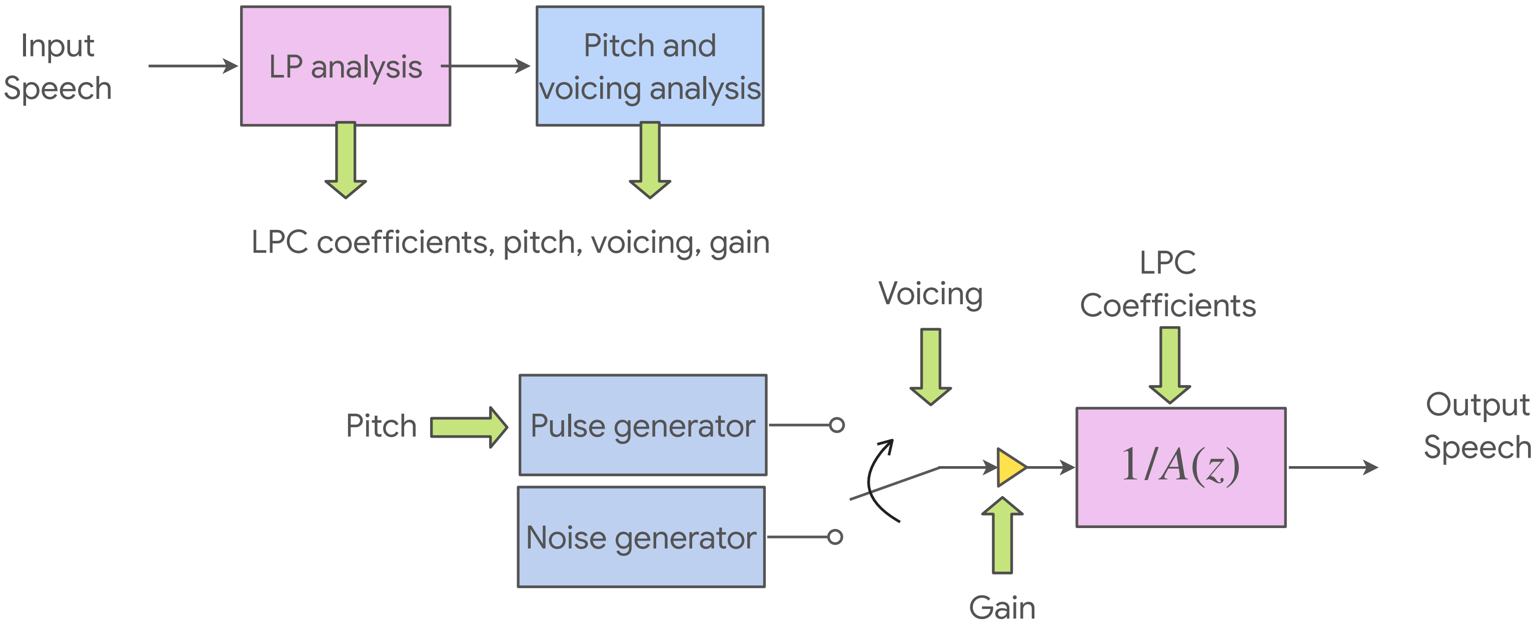
$$s(n) = u(n) + \sum_{k=1}^{N} a_k s(n-k)$$

Polynomial coefficients $\mathbf{a} = [1, a_1, a_2, \ldots, a_N]$

Solution to the normal equation $\mathbf{a} = \mathbf{R}_{ss}^{-1}\mathbf{r}_{ss}$

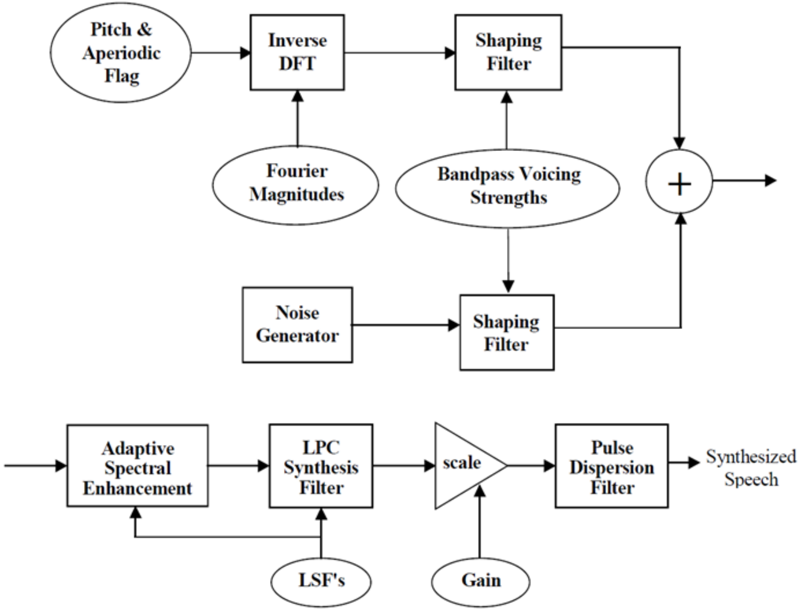
Correlation matrix and vector containing $r_{ss}(|i-j|) = \sum_{n} s[n-i]s[n-j]$

LPC coding


Input
Speech
LP analysis
Pitch and
voicing analysis
LPC coefficients, pitch, voicing, gain
Voicing
LPC
Coefficients
Pitch
Pulse generator
Noise generator
$1/A(z)$
Output
Speech
Gain

MELP

Source-filter vocoding in the time domain

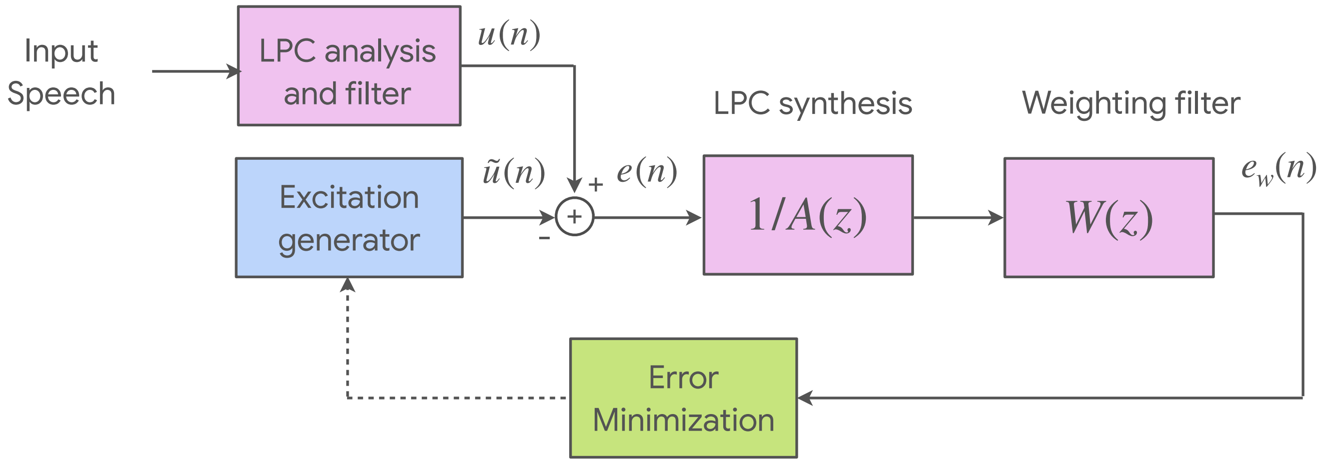


- Originally by McCree and Barnwell, 1995 [3]
- US federal standard in 1996
- Operates at 600, 1200, or 2400 bps

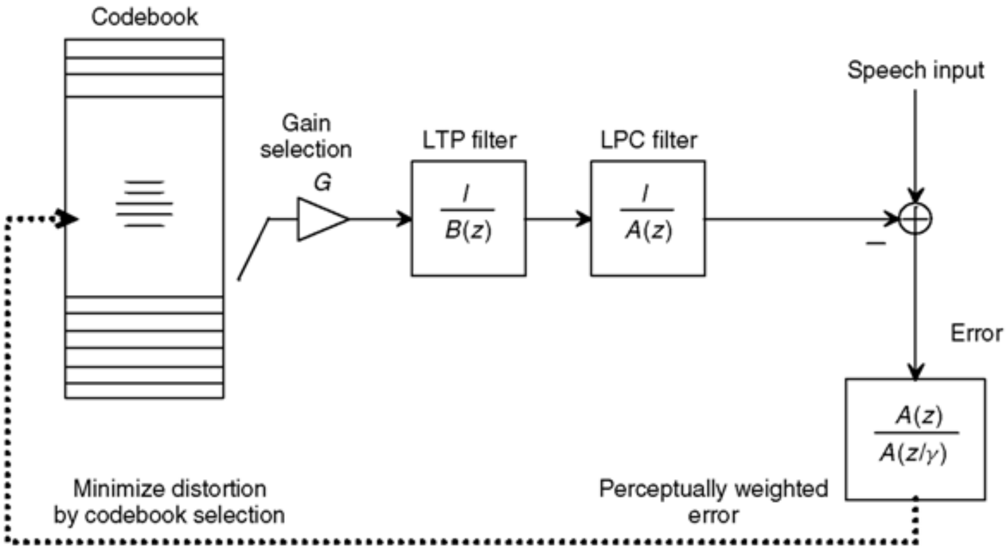
[3] *A. V. McCree and T. P. Barnwell*, "A mixed excitation LPC vocoder model for low bit rate speech coding," in *IEEE Trans. Speech and Audio Proc., vol. 3, no. 4, pp. 242-250, 1995*

Linear predictive analysis-by-synthesis coding

Waveform-matching (“hybrid”) coding



CELP



- Introduced by Schroeder and Atal, 1985 [4]
- In most digital telephony standards
- Bit rates from ~4 kbps to ~25 kbps

[4] *M. R. Schroeder and B. S. Atal*, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates", *ICASSP 1985*

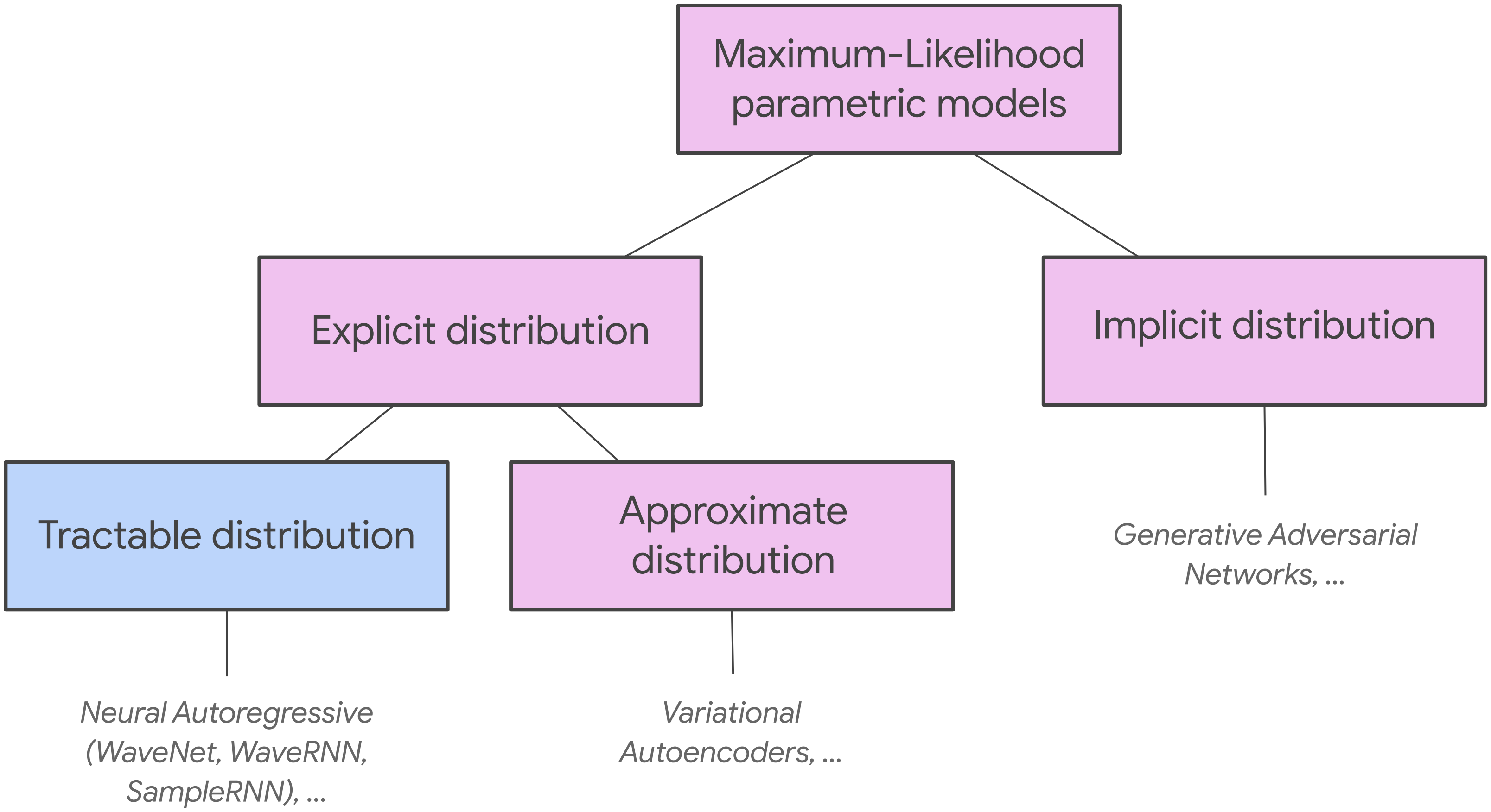
Speech coding state for VoIP calls and conferencing around 2017

- Typical rates for internet calls and conferencing, 16 - 32 kbps (wideband)
- Codecs in use are waveform-matching coders - poor at low rates.
- Reducing bit rate becoming important for poor networks, e.g., emerging markets.
- Parametric coders operate at lower than 5 kbps, but suffer from poor quality from the synthesis models.
- Wait, parametric coders require a **generative model** at the decoder

Can we do better?

- Source-filter modeling for speech synthesis
- Speech coding
- Linear predictive coding
- **Generative neural synthesis for coding**
- LPCNet
- Noise-robust neural vocoding

Generative models taxonomy



Modelling data distribution

- Let data $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$, and we're given a finite set of samples from this distribution
  $$\mathcal{X} = \{\mathbf{x} : \mathbf{x} \sim p_{\text{data}}(\mathbf{x})\}, |\mathcal{X}| = N$$
- We want to find a model such that $p_{\text{data}}(\mathbf{x}) \approx p_{\text{model}}(\mathbf{x}; \theta)$
- Man-made parametric models (mixtures of Gaussians, Laplacian, Weibull, Poisson, etc.) are limited in expressivity
- Modern deep models remove this issue

Neural autoregressive models

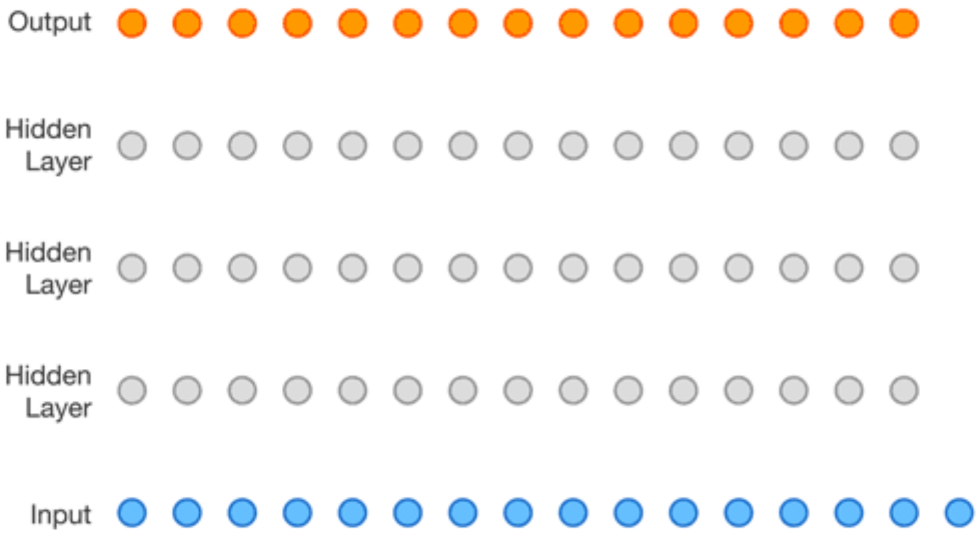
- Again, $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$
- Since it's a vector, we can factorize per dimension

$$p_{\text{data}}(\mathbf{x}) = q(x_1)q(x_2|x_1)q(x_3|x_2,x_1)\cdots q(x_d|x_{d-1},x_{d-2},\ldots,x_1)$$

$$p_{\text{data}}(\mathbf{x}) = q(x_1)\prod_{k=2}^{d} q(x_k|x_{k-1},x_{k-2},\ldots,x_1)$$

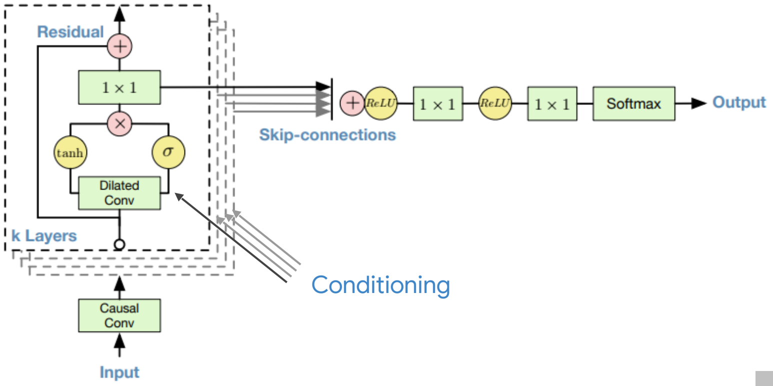
- For audio this means we can predict next sample given previous samples
- Autoregressive models admit a tractable and explicit likelihood, and can
  - Draw a sample
  - Assign a probability to a sample

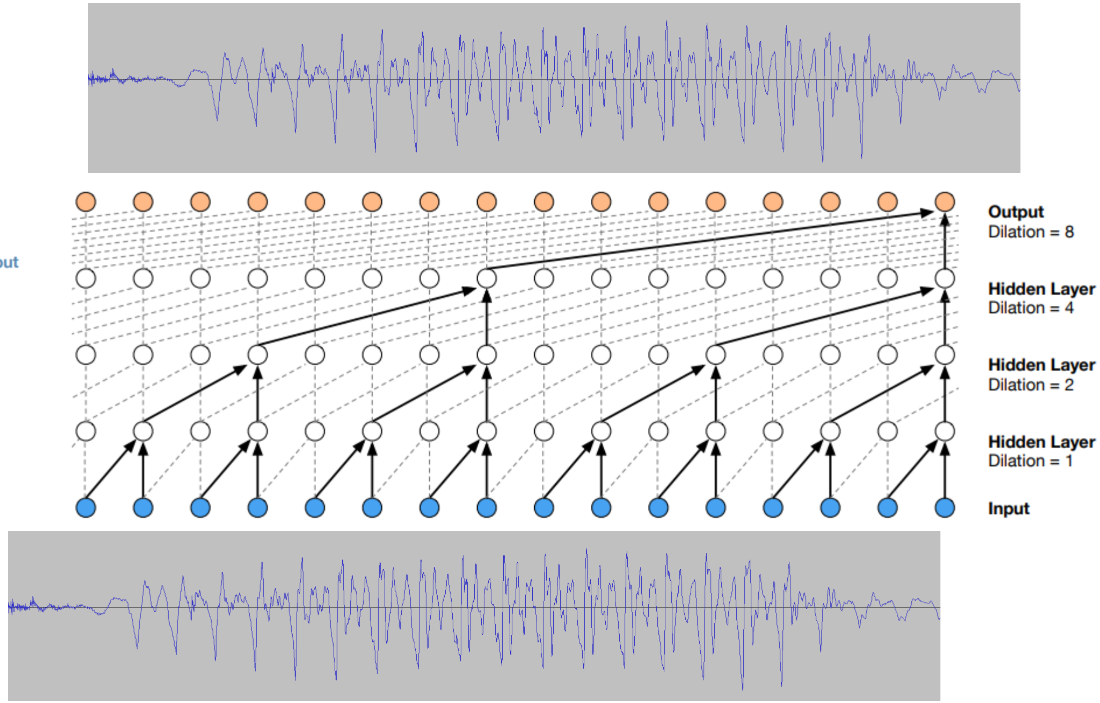
What's WaveNet?



[5] *A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K Kavukcuoglu,* "WaveNet: A Generative Model for Raw Audio," *ArXiv:1609.03499, 2016*

WaveNet architecture





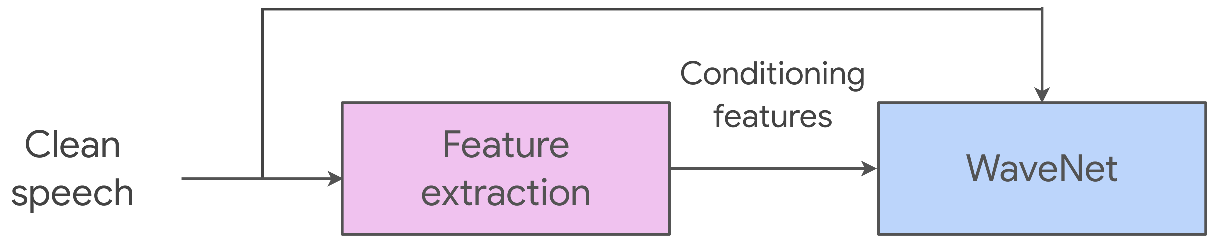
- Needs conditioning to avoid babble

A WaveNet-based parametric codec [6]

[6] *W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters,*
"Wavenet based low rate speech coding," *ICASSP 2018*

Parametric WaveNet

Training

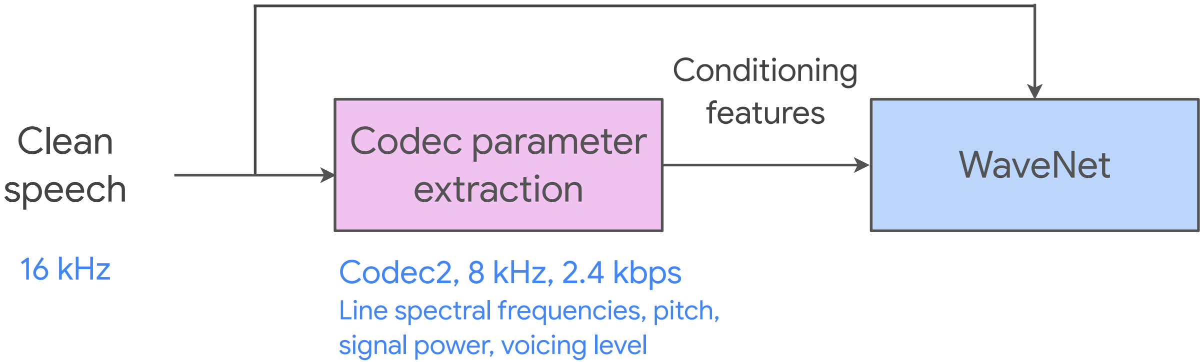


Condition the generative model:

$p(\mathrm{x}|\theta)$

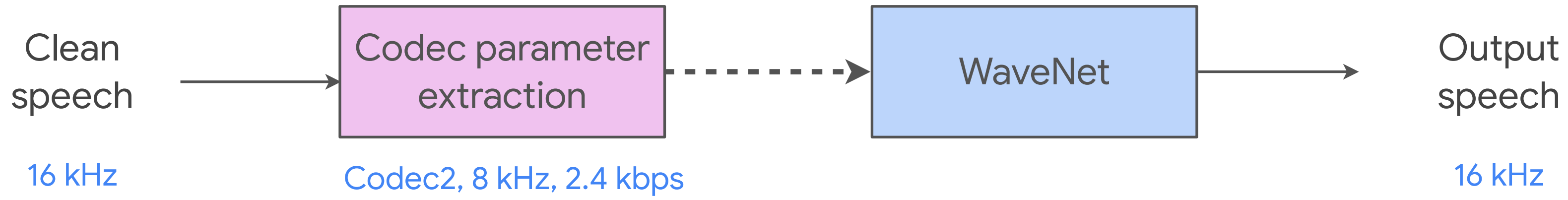
Parametric WaveNet

Training



Parametric WaveNet coding

Codec operation



Experiment setup

- Vocoder params extracted with Codec2 @ 2.4 kbps
- Input features: 8 kHz
- Target output speech: 16 kHz
- Dataset: WSJ0

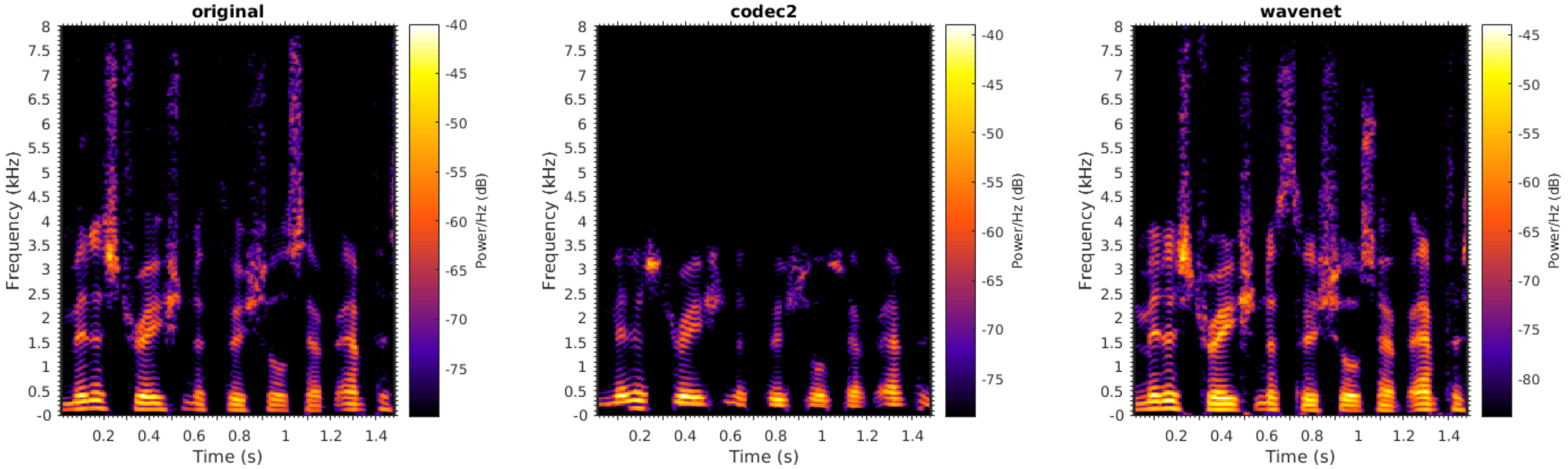
    Training: 32580 utterances, 123 speakers

    Test: 2907 utterances, 8 speakers
- Standard 8-bit μ-law WaveNet model used

    Conditional variables updated at 100 Hz

    Receptive field: ~300 ms

Bandwidth extension!



Quality

POLQA mean opinion scores
(Rates are in kbps)

	Codec 2	*MELP*	*Speex*	*AMR-WB*	*WW*	*WP*
Rate	2.4	2.4	2.4	23	42	2.4
MOS	2.7	2.9	2.2	4.6	4.7	~~2.9~~

- Conventional objective quality measures are not useful
- The parametric WaveNet coder generates a likely waveform, rather than reproduce the signal

Listening tests: Mushra-esque

Demo

Practical speech coding requirements

- **Sufficiently low complexity (original WaveNet infeasible)**
  - Parallel/distilled WaveNet [7]
  - WaveRNN [8]
  - SampleRNN [9]
- **Robustness to diverse conditions**
  - Background noise (examples)
  - Recording chain (hardware, processing)
  - Multiple talkers and languages

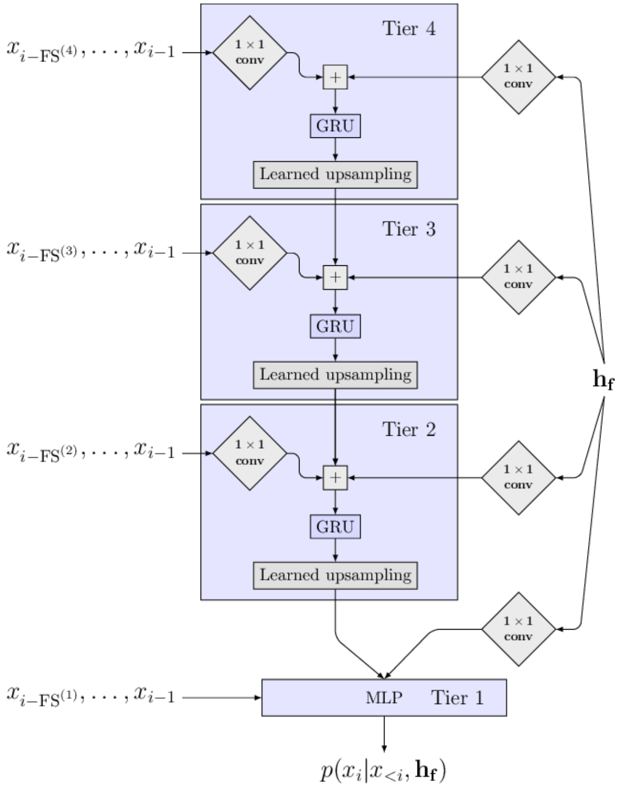
[7] *A. van der Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van der Driessche, E. Lockhart, L. C. Cobo, F. Stimberg,* "Parallel WaveNet: Fast high-fidelity speech synthesis, " *preprint arXiv:1711.10433, 2017*

[8] *N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van der Oord, S. Dieleman, K. Kavukcuoglu,* "Efficient neural audio synthesis", *preprint arXiv:1802.08435, 2018*

[9] *S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, Y. Bengio,* "SampleRNN: An unconditional end-to-end neural audio generation model," *preprint arXiv:1612.07837, 2016*

SampleRNN for coding[10]

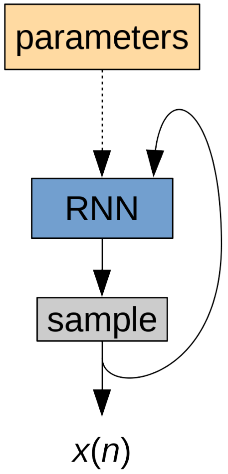
- Lower complexity than WaveNet
- 3-layer hierarchical GRUs at different time scales
- Conditioned on LPC vocoder parameters



[10] *J. Klejsa, P. Hedelin, C. Zhou, R. Fejgin, L. Villemoes,* "High-quality speech coding with SampleRNN," *ICASSP 2019*

WaveRNN

- Single layer RNN (GRU)
- Sparse weight matrices
- Coarse and fine parts for 16 bit resolution



Update equations, omitting fine resolution

$$\mathbf{x}_t = \left[s_{t-1}; \mathbf{f}\right]$$

$$\mathbf{u}_t = \sigma\left(\mathbf{W}^{(u)}\mathbf{h}_{t-1} + \mathbf{U}^{(u)}\mathbf{x}_t\right)$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}^{(r)}\mathbf{h}_{t-1} + \mathbf{U}^{(r)}\mathbf{x}_t\right)$$

$$\widetilde{\mathbf{h}}_t = \tanh\left(\mathbf{r}_t \circ \left(\mathbf{W}^{(h)}\mathbf{h}_{t-1}\right) + \mathbf{U}^{(h)}\mathbf{x}_t\right)$$

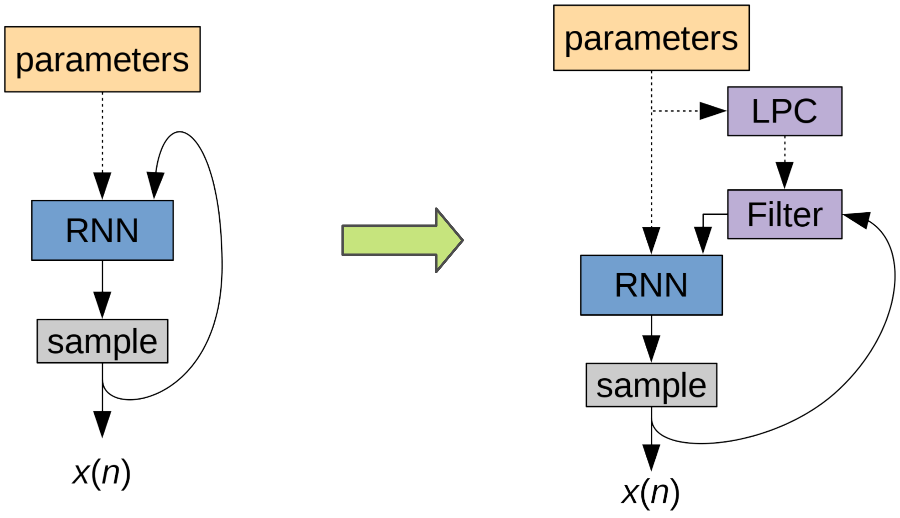
$$\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + \left(1 - \mathbf{u}_t\right) \circ \widetilde{\mathbf{h}}_t$$

$$P\left(s_t\right) = \mathrm{softmax}\left(\mathbf{W}_2\ \mathrm{relu}\left(\mathbf{W}_1\mathbf{h}_t\right)\right)$$

- Source-filter modeling for speech synthesis
- Speech coding
- Linear predictive coding
- Generative neural synthesis for coding
- **LPCNet**
- Noise-robust neural vocoding

LPCNet[11]

Let the network generate excitation



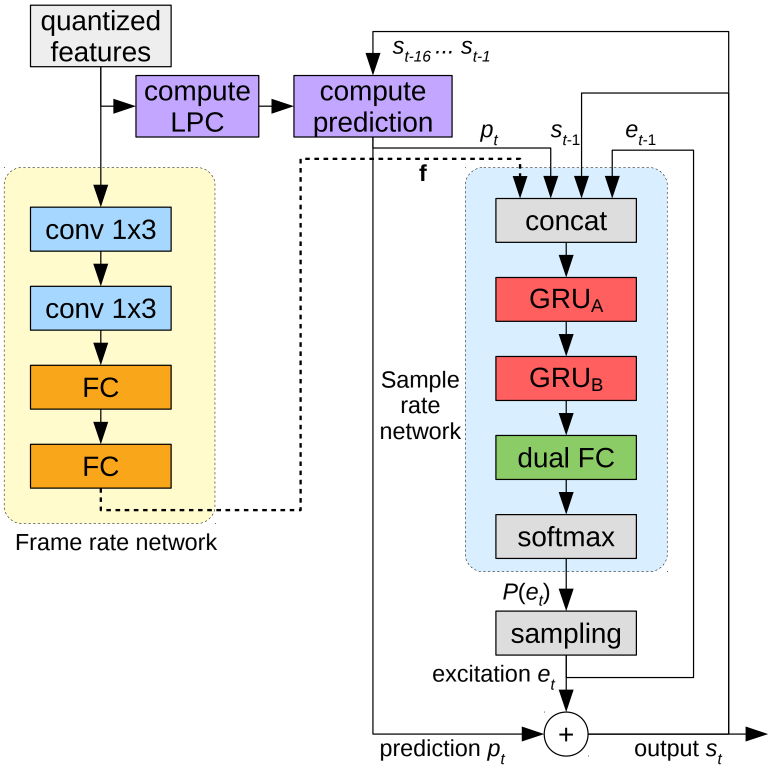
[11] *J.-M. Valin, J. Skoglund,* “LPCNet: Improving neural speech synthesis through linear prediction,” *ICASSP 2019*

LPCNet
Other improvements

- **Pre-emphasis**
  - Boost HF in input/training data
  - Apply de-emphasis on synthesis
  - Reduce perceived noise in wideband

- **Input embedding**
  - Rather than u-law values directly, consider as one-hot classifications
  - Learning non-linear functions
  - No extra cost by pre-computing matrix products

LPCNet synthesis



Update equations

$$\mathbf{x}_t = \left[s_{t-1}; \mathbf{f}\right]$$

$$\mathbf{u}_t = \sigma\left(\mathbf{W}_u \mathbf{h}_{t-1} + \mathbf{v}^{(u,s)}_{s_{t-1}} + \mathbf{v}^{(u,p)}_{p_t} + \mathbf{v}^{(u,e)}_{e_{t-1}} + \mathbf{g}^{(u)}\right)$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \mathbf{h}_{t-1} + \mathbf{v}^{(r,s)}_{s_{t-1}} + \mathbf{v}^{(r,p)}_{p_t} + \mathbf{v}^{(r,e)}_{e_{t-1}} + \mathbf{g}^{(r)}\right)$$

$$\widetilde{\mathbf{h}}_t = \tau\left(\mathbf{r}_t \circ \left(\mathbf{W}_h \mathbf{h}_{t-1}\right) + \mathbf{v}^{(h,s)}_{s_{t-1}} + \mathbf{v}^{(h,p)}_{p_t} + \mathbf{v}^{(h,e)}_{e_{t-1}} + \mathbf{g}^{(h)}\right)$$

$$\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + \left(1 - \mathbf{u}_t\right) \circ \widetilde{\mathbf{h}}_t$$

$$P\left(e_t\right) = \operatorname{softmax}\left(\operatorname{dual\_fc}\left(\operatorname{GRU_B}\left(\mathbf{h}_t\right)\right)\right)$$

$$\operatorname{dual\_fc}(\mathbf{x}) = \mathbf{a}_1 \circ \tau\left(\mathbf{W}_1 \mathbf{x}\right) + \mathbf{a}_2 \circ \tau\left(\mathbf{W}_2 \mathbf{x}\right)$$

LPCNet for coding[12]

Speech features

- **Conditioning features: 10 ms**
  - Cepstrum
  - Pitch period
  - Pitch correlation
- **Packets: 40 ms**
  - Packing 4 frames

[12] *J.-M. Valin, J. Skoglund,* “A Real-Time Wideband Neural Vocoder at 1.6 kb/s Using LPCNet,” *Interspeech 2019*

LPCNet for coding

Pitch

- **Detection**
  - Cross-correlation on LPC residual
  - 5 ms sub-rame
  - Range 62.5 Hz to 500 Hz
- **Quantization**
  - Log-scale pitch over packet (6 bits)
  - Linear pitch modulation (3 bits)
  - Pitch correlation (2 bits)

LPCNet for coding

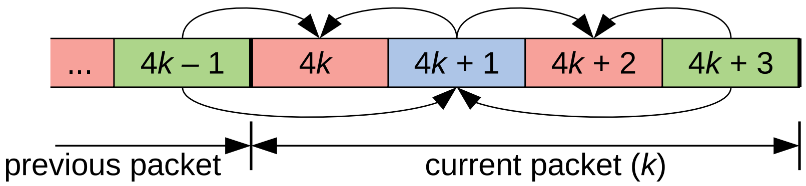
Cepstrum

- **Cepstral coefficients over 18 Bark bands**
  20-ms windows (50% overlap)
- **Quantization using two-way prediction**
  Past sub-frame, future sub-frame, or average

Prediction, no VQ (3 bits for both vectors)

Independent 3-stage VQ (30 bits)

Prediction + VQ (13 bits)



LPCNet for coding
Bit allocation

Parameter	Bits
Pitch period	6
Pitch modulation	3
Pitch correlation	2
Energy (C0)	7
Cepstrum VQ (40ms)	30
Cepstrum delta (20 ms)	13
Cepstrum interpolation (10 ms)	3
Total	**64**

Training

- **Add noise to input to reduce effects of teacher forcing**
- **Two-step training**

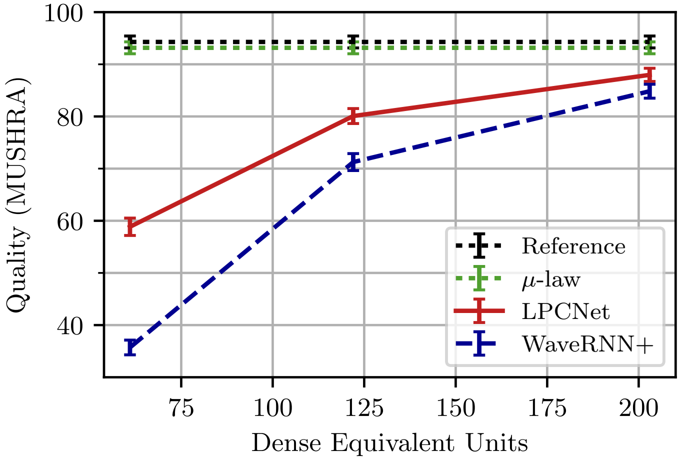
  Network trained with unquantized features

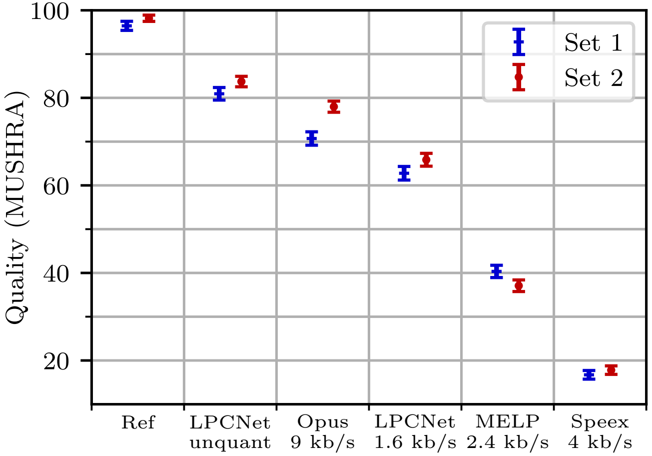
  Frame rate network adapted with quantized features (sample rate network frozen)

LPCNet complexity

CPU	Clock	% Core
*AMD 2990WX (Threadripper)	3.0 GHz	14%
*Xeon E5-2640 v4 (Broadwell)	2.4 GHz	20%
Snapdragon 855 (Galaxy S10)	2.8 GHz	31%
Snapdragon 845 (Pixel 3)	2.5 GHz	68%
Cortex-A72 (Raspberry Pi 4)	1.5 GHz	110%
*turbo enabled		

LPCNet speech quality





MUSHRA-like listening tests with 100 crowd-sourced raters

Set 1: NTT database (similar but disjunct from training set)

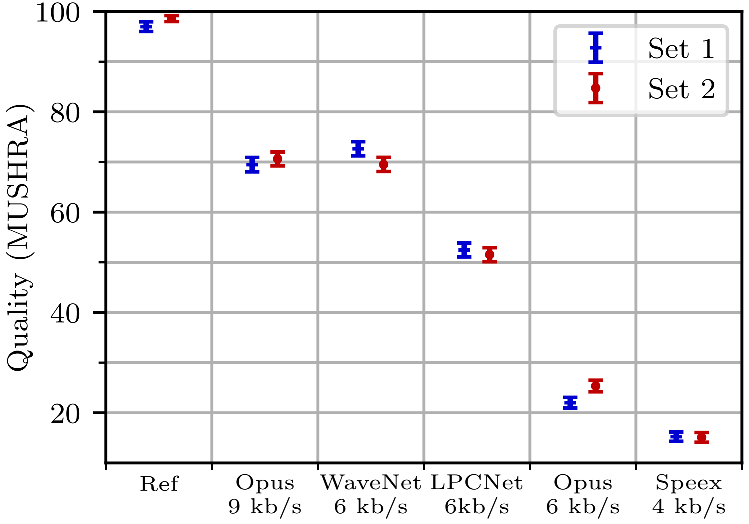
Set 2: Opus standard test vectors

LPCNet as synthesis of Opus decoder [13]

- **Opus codec**
  - IETF-standardized speech and audio codec
  - Supports narrowband to fullbandCombination of LPC-based SILK and transform-based CEL
  - Focus on wideband speech in SILK
  - Waveform-matching codec
- **Conditioning features from decoded Opus bit stream**
  - Spectral features from both bit stream and decoded audio
  - Two pitch parameters, period and average gain
- **Comparing with WaveNet as generative synthesis**
  - As an unimplementable upper limit

[13] *J. Skoglund, J.-M. Valin,* "A Real-Time Wideband Neural Vocoder at 1.6 kb/s Using LPCNet," *Interspeech 2020*

Speech quality of post processing



Set 1: NTT database (similar but disjunct from training set)

Set 2: Opus standard test vectors

MUSHRA-like listening tests with 100 crowd-sourced raters

- Source-filter modeling for speech synthesis
- Speech coding
- Linear predictive coding
- Generative neural synthesis for coding
- LPCNet
- **Noise-robust neural speech coding**

Addressing background noise in neural vocoding[14]

- Focusing on robustness to noisy input
- Disregarding complexity

[14] *F. S. C. Lim, W. B. Kleijn, M. Chinen, J. Skoglund,* "Robust low rate speech coding based on cloned networks and WaveNet," *Interspeech 2020*

Proposed system

- **Hypothesis**

  Generative models perform best when synthesizing signals from a single source

- **Proposed codec**

  1. Extract speech features from a noisy input
  2. Quantize features for transmission
  3. Use as conditioning features to WaveNet to synthesize the clean output speech

Extract speech features: clone-based training

An input set of **perceptually equivalent speech signals**

{

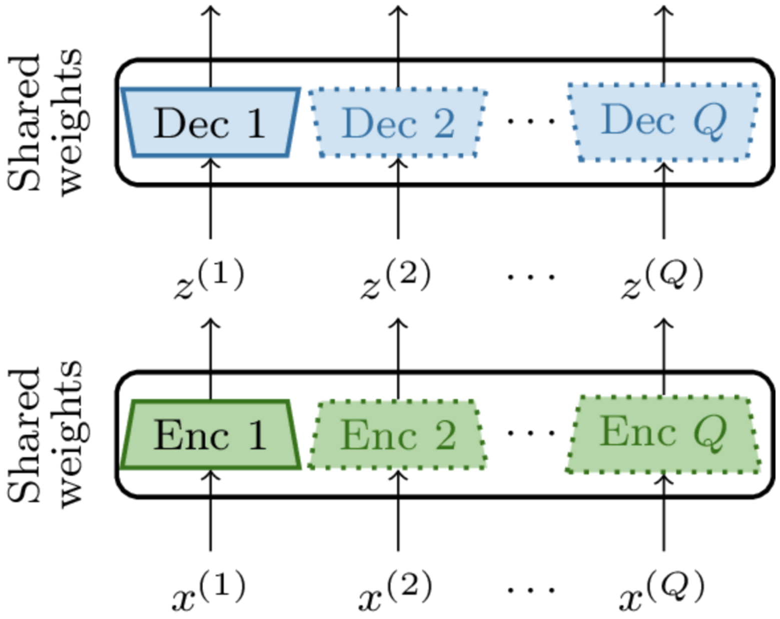
“The birch canoe slid on the smooth planks”,

“The birch canoe slid on the smooth planks” **+ car noise**,

“The birch canoe slid on the smooth planks” **+ kitchen noise**,

}

Extract speech features: clone-based training



$x^{(1)}$ = “The birch canoe...”

$x^{(2)}$ = “The birch canoe...” + car noise

$x^{(Q)}$ = “The birch canoe ...” + kitchen noise

Extract speech features: clone-based training

TRAINING LOSSES

1. **Equivalent input signals map to identical features**

$$F_1 = \sum_{q=2}^{Q} \|z^{(1)} - z^{(q)}\|_2$$

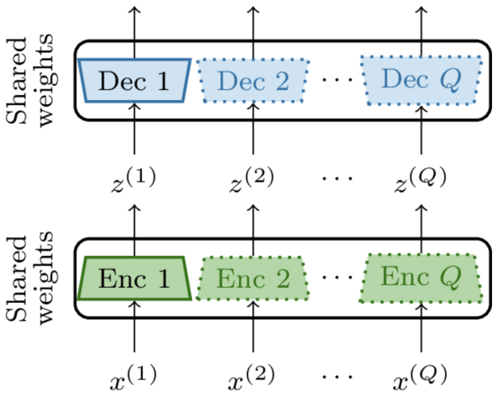
2. **Latent features are distributed as a factorized Laplacian distribution**

   Encourages independence (disentanglement) of the features

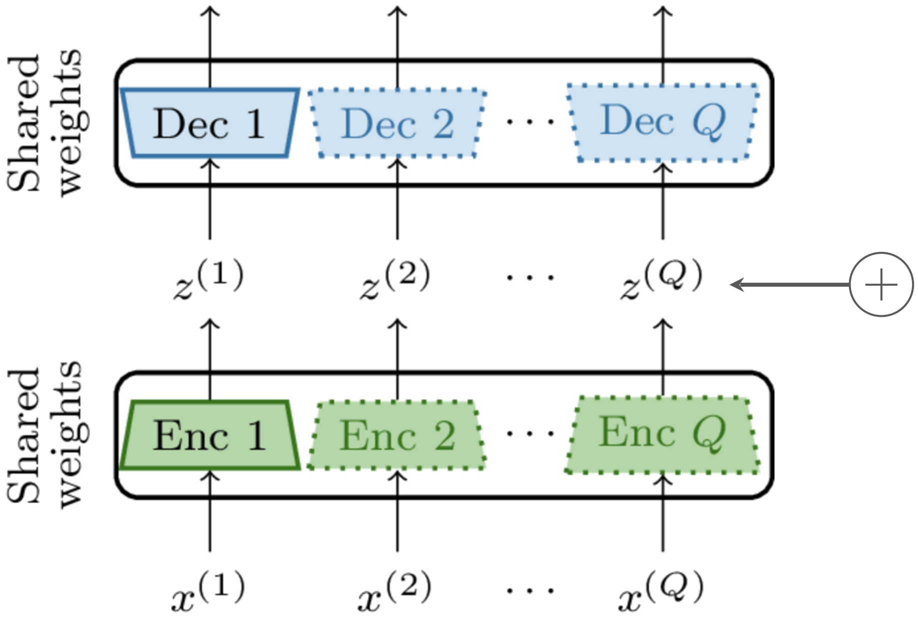
   Use the maximum mean discrepancy loss over the batch, per feature

3. **Decoded features match the features from the clean input**

$$F_3 = \sum_{q=1}^{Q} \|x - \mathrm{Dec}(z^{(q)})\|_2$$



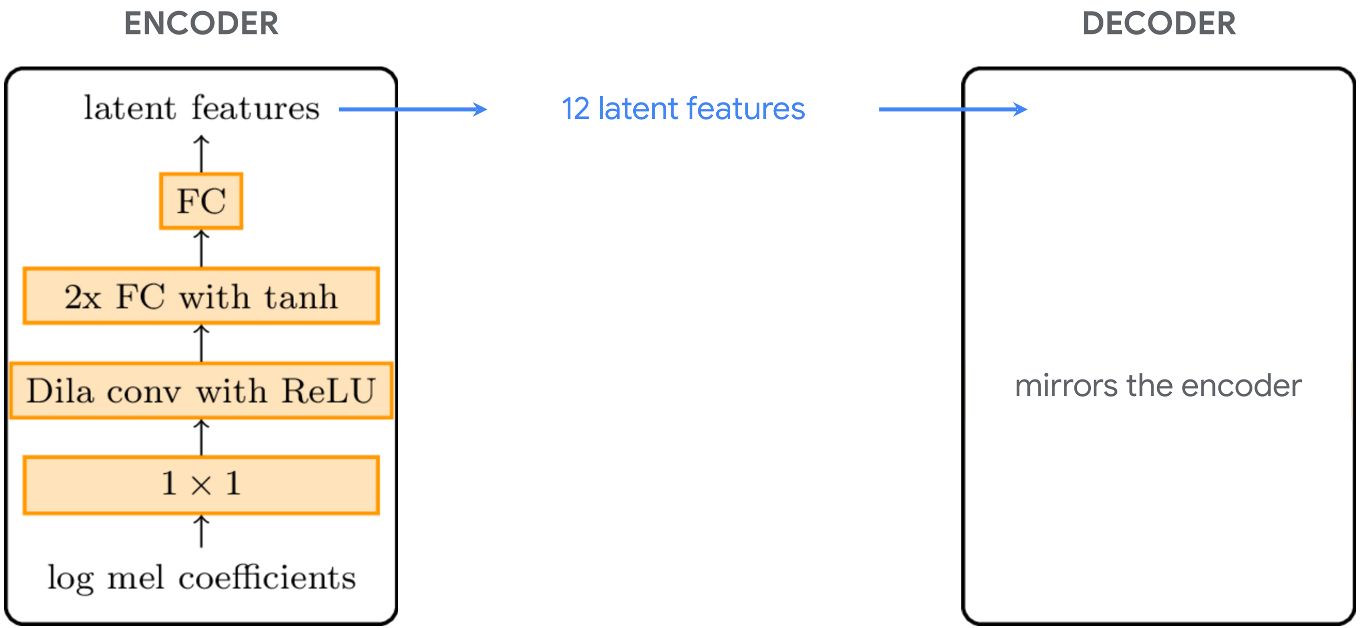
Extract speech features: clone-based training



(at training only)

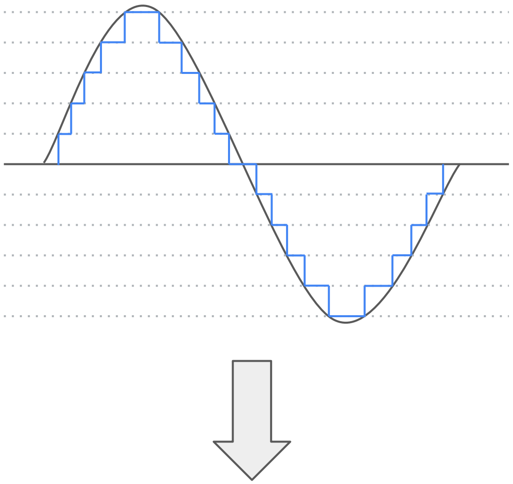
Gaussian noise

Extract speech features: clone-based training



Quantize speech features

UNIFORM SCALAR QUANTIZER



ENTROPY CODING

Per-dimension entropy upper bound

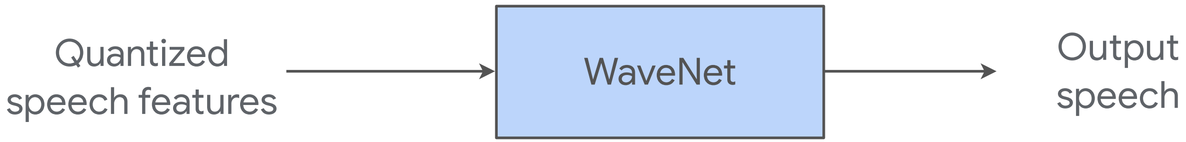
feature variance

$$H(I) \leq \frac{1}{2} \log \left(\frac{2\pi e \sigma^2}{\Delta^2} \right)$$

target bitrate

step size?

Synthesize output speech



- Use the original WaveNet model
- Replace 8-bit softmax output layer with 16-bit discretized logistic mixtures [15]

[15] *Salimans, A. Karpathy, X. Chen, and D. P. Kingma,* "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," *in Proc. ICLR, 2017*

Experimental setup

Dataset

- WSJ0 and LibriTTS @ 16 kHz
- Training dataset: ~255 hours, ~1k speakers
- Test dataset: ~16 hours, 47 speakers
- Additive noise from Freesound dataset and internal recordings (cafes, busy streets, offices etc.)

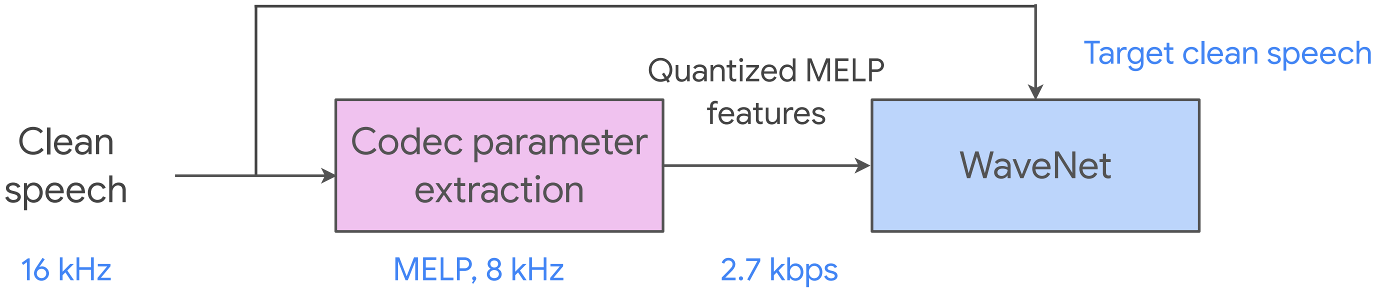
Clone-based Feature Extractor

- 8 clones: 1 clean speech and 7 noisy versions
- SNR = 0 to 30 dB

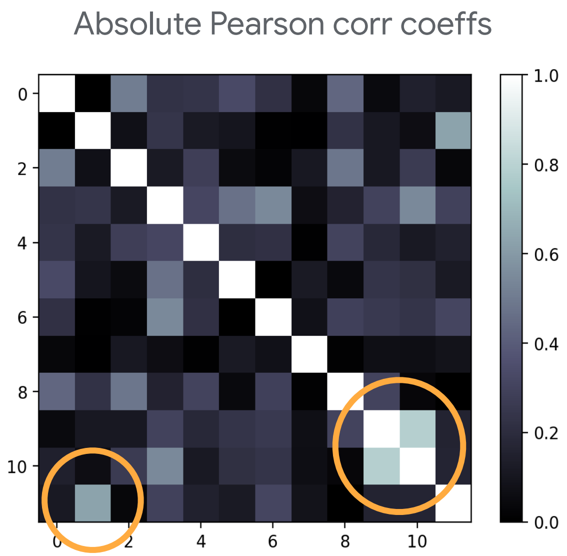
Quantizer

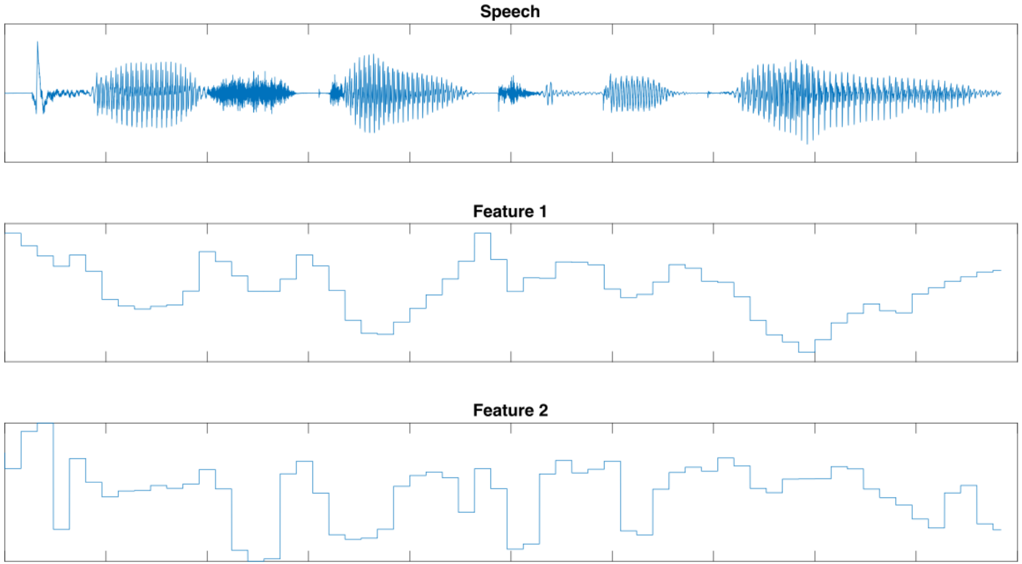
- Total target bitrate: 2 kbps
- Actual entropy computed: ~37 bits / frame = ~1.8 kbps

Reference system: MELP WaveNet

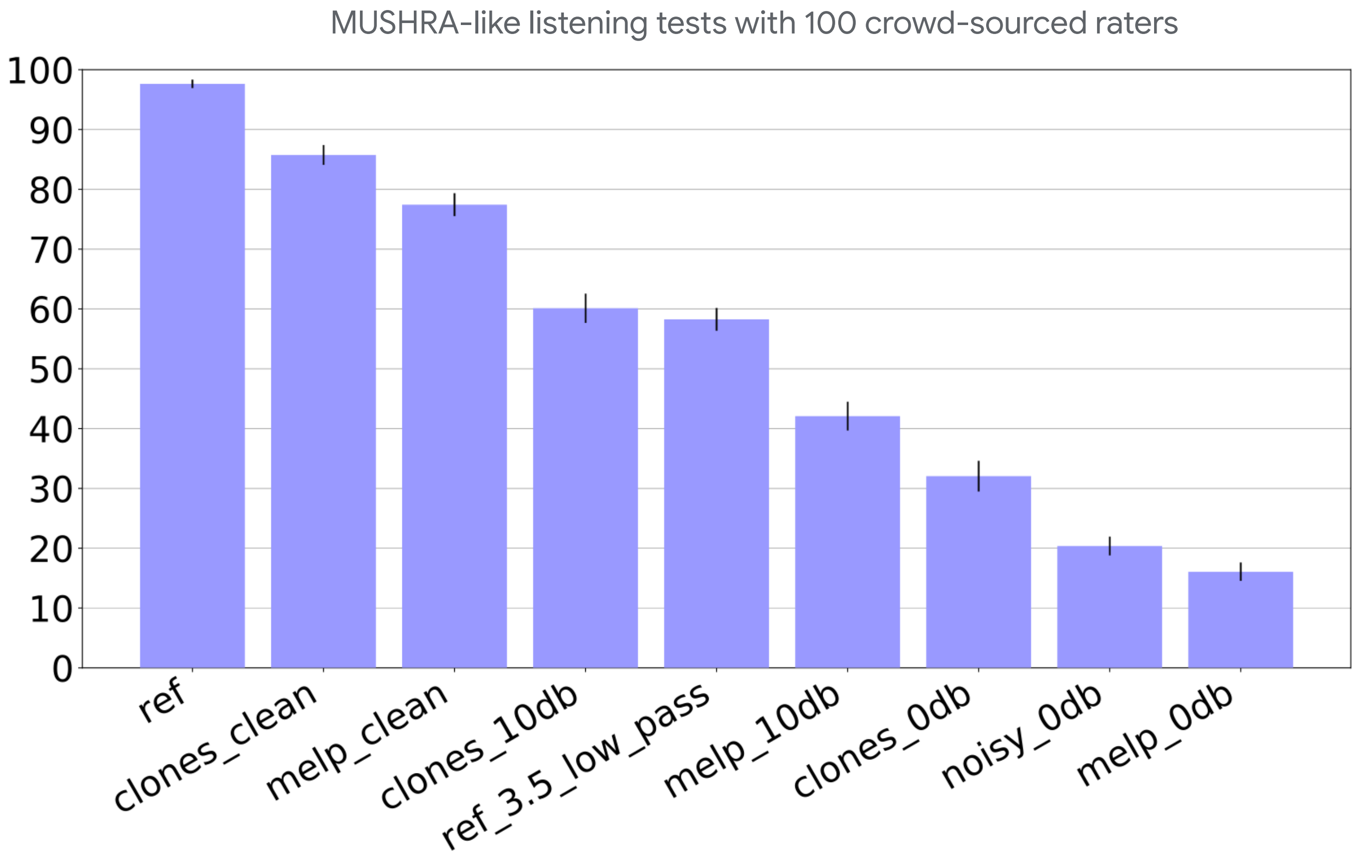


Inspecting the non-quantized latent features

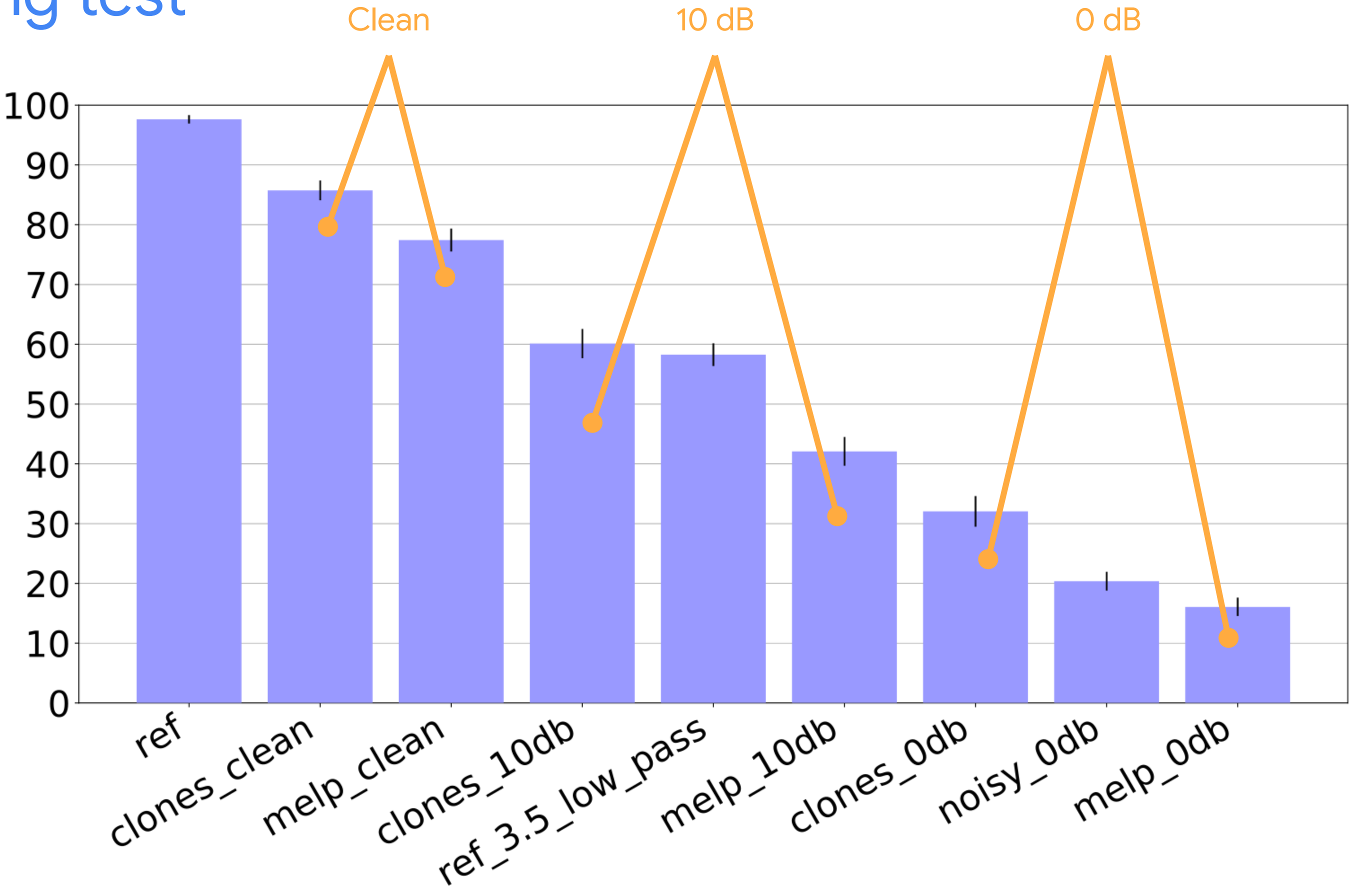




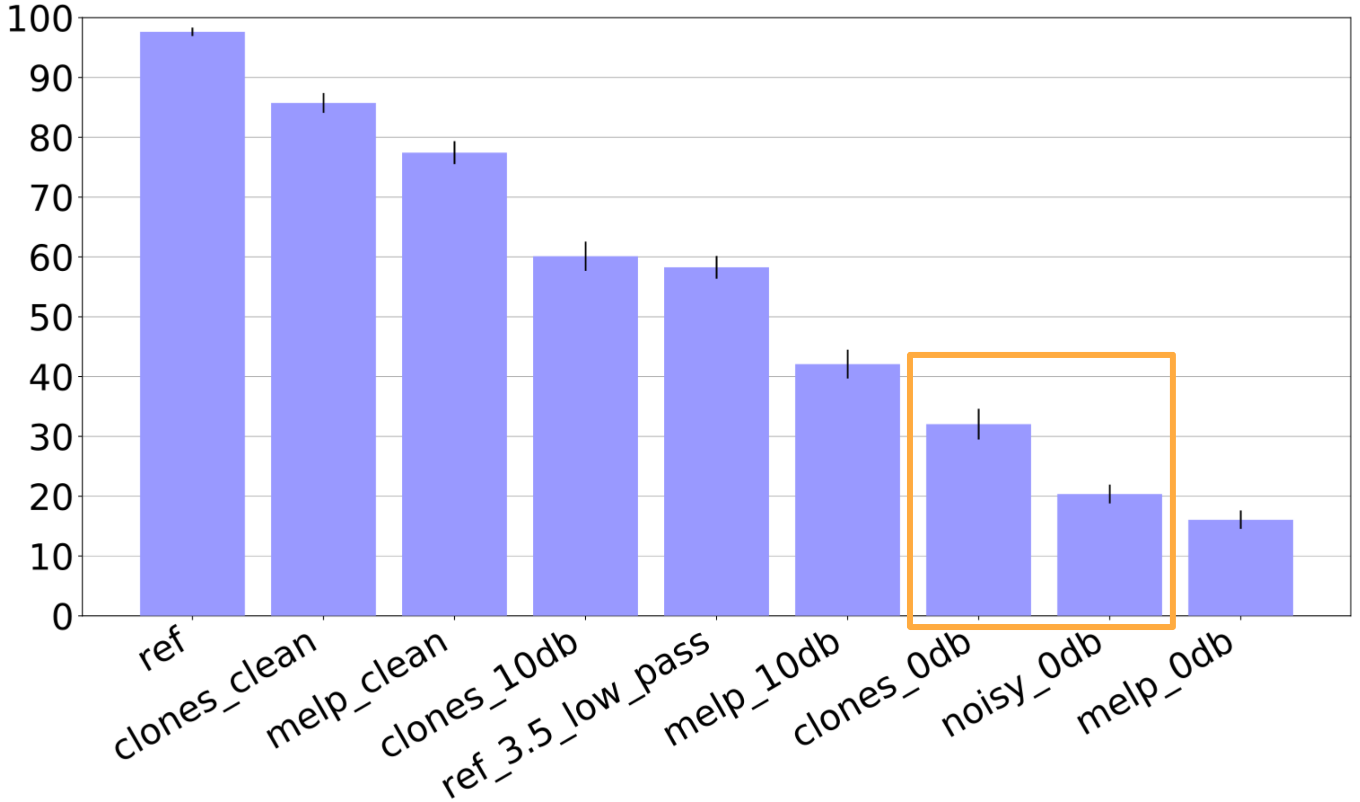
Listening test



Listening test



Listening test



Deep Generative Models for Speech Compression

Thank you for listening!

Jan Skoglund - Chrome Media Audio