Improving LPCNet-based Text-to-Speech with Linear Prediction-structured Mixture Density Network

Min-Jae Hwang[1,2], Eunwoo Song[3], Ryuichi Yamamoto[4],
Frank Soong[5] and Hong-Goo Kang[2]

[1]Search Solutions Inc., South Korea
[2]Yonsei Univ., South Korea, [3]NAVER Corp., South Korea
[4]Line Corp., Japan, [5]Microsoft Research Asia, China

Overview

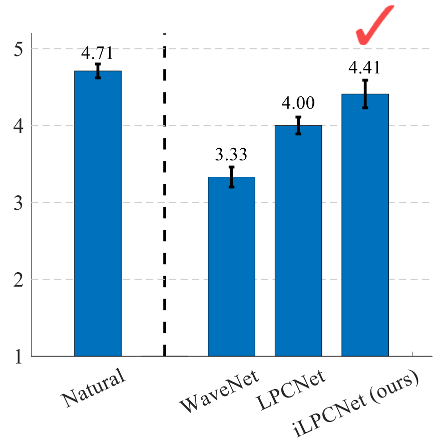
Paper objective

- Improving the quality of LPCNet-based parametric speech synthesis system

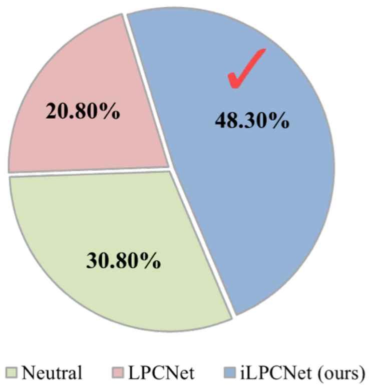
Proposed systems

- LP-MDN: Linear prediction-structured mixture density network
  - Structurally merge the LP process with an autoregressive neural vocoding framework
- iLPCNet: Improved LPCNet vocoder
  - Incorporating LP-MDN into LPCNet framework
- Effective training and generation methods

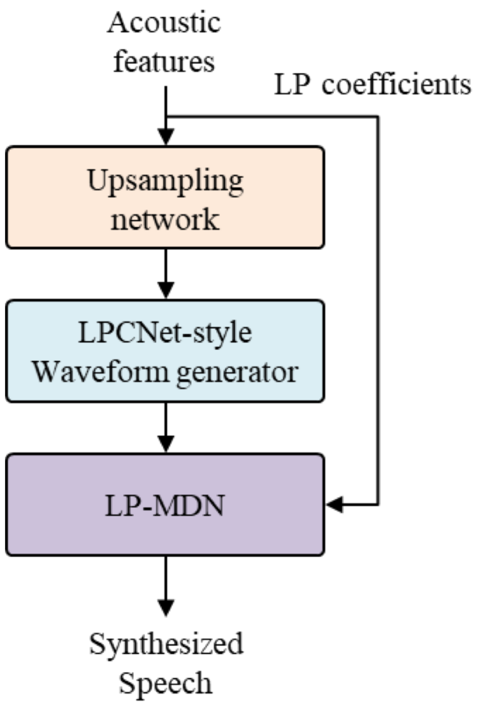
Performance



[MOS test result]



[Preference test result]



[Overview of proposed iLPCNet]

CONTENTS

[1] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019.

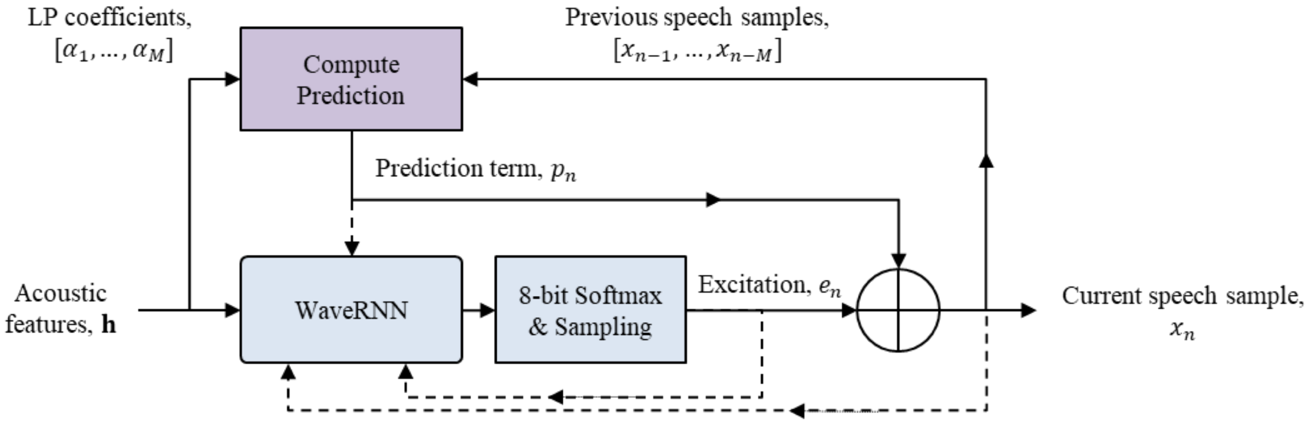
LPCNet-based neural vocoding

[LP synthesis process]

$$p_n = \sum_{i=1}^{P} \alpha_i x_{n-i}.$$

$$x_n = e_n + p_n$$

Incorporate linear prediction (LP) structure within WaveRNN framework



[Block diagram of LPCNet]

Characteristics

- WaveRNN architecture
  - Accelerate the generation speed of autoregressive neural vocoder
- LP synthesis-based spectral shaping filter
  - Achieve good synthesis quality by attenuating quantization noise caused by μ-law modeling
- Various tuning methods for μ-law modeling
  - Waveform embedding, discrete training noise injection, conditional sampling for softmax distribution, pre-emphasis filter, …

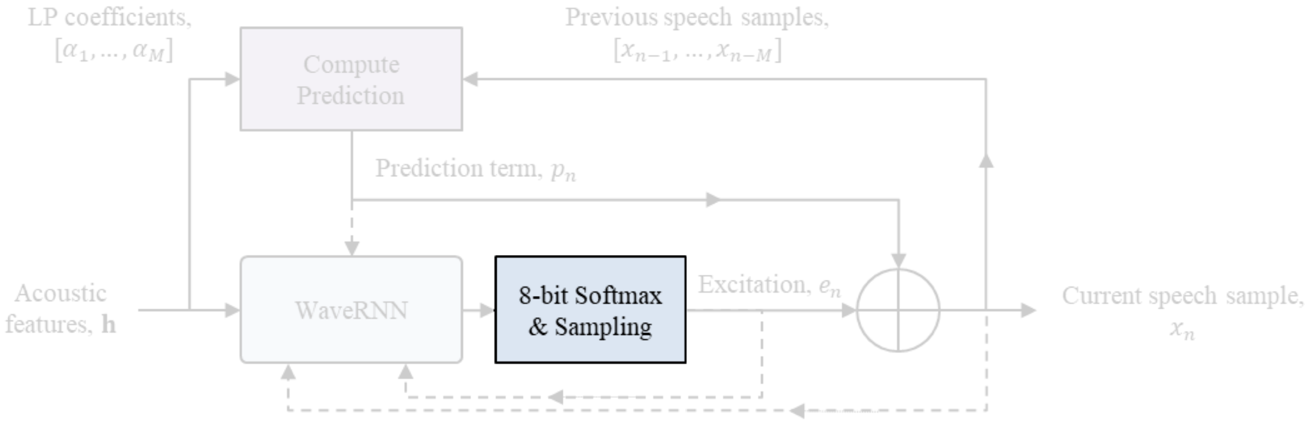
LPCNet-based neural vocoding

[LP synthesis process]

$$p_n = \sum_{i=1}^{P} \alpha_i x_{n-i}.$$

$$x_n = e_n + p_n$$

Incorporate linear prediction (LP) structure within WaveRNN framework



[Block diagram of LPCNet]

Methods to improve performance

- Replace the μ-law waveform model with a continuous waveform model
  - Improve synthesis quality by utilizing densely distributed waveform sample
  - Simplify the tuning methods

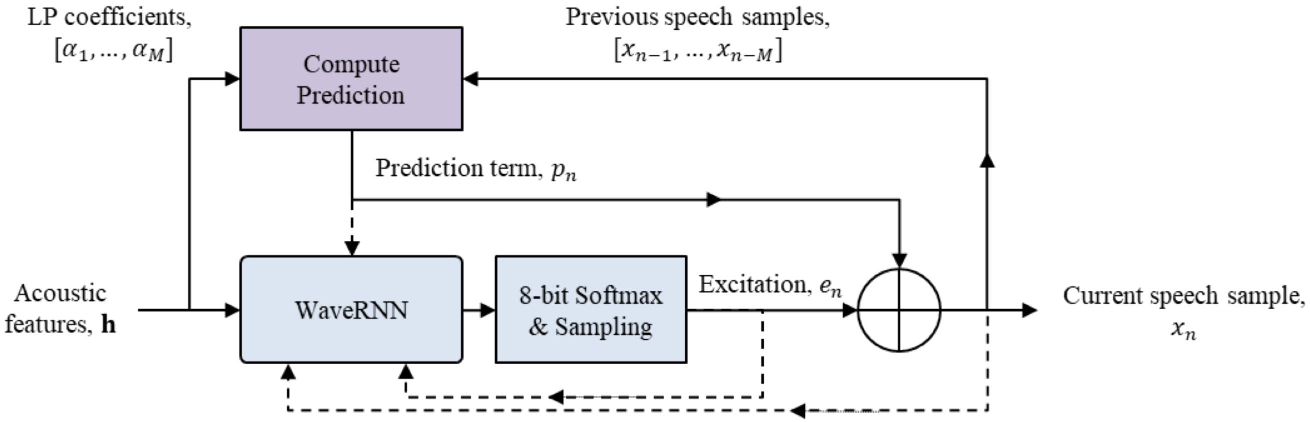
LPCNet-based neural vocoding

[LP synthesis process]

$$p_n = \sum_{i=1}^{P} \alpha_i x_{n-i}.$$

$$x_n = e_n + p_n$$

Incorporate linear prediction (LP) structure within WaveRNN framework



[Block diagram of LPCNet]

Methods to improve performance

- Replace the μ-law waveform model with a continuous waveform model
  - Improve synthesis quality by utilizing densely distributed waveform sample
  - Simplify the tuning methods
- Suggest a closed-loop solution of LP structure for compact representation

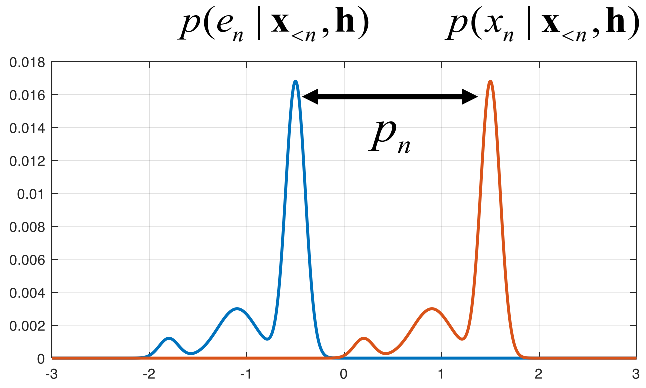
Linear prediction Mixture density network

LP-STRUCTURED MDN

Basic assumption on autoregressive neural vocoder

1. Previous speech samples, $\mathbf{x}_{<n}$, are given
2. LP coefficients, $\{\alpha_{n,i}\}$, are given

Their linear combination, $p_n = \sum_{i=1}^{P} \alpha_{n,i} x_{n-i}$ *, are also given*



[Conditional distributions of speech and excitation]

Probabilistic analysis

$$x_n = e_n + p_n$$

$$X_n \mid (\mathbf{x}_{<n}, \mathbf{h}) = E_n \mid (\mathbf{x}_{<n}, \mathbf{h}) + p_n$$

Random variables X_n and E_n are depends on only the constant difference of p_n

Mixture of Gaussian (MoG) modeling

$$p(x_n \mid \mathbf{x}_{<n}, \mathbf{h}_n) = \sum_{n=1}^{N} \omega_n \cdot \frac{1}{\sqrt{2\pi} s_{n,i}} \exp\left[\frac{(x_n - \mu_{n,i})^2}{2 s_{n,i}^2}\right]$$

- Utilize the shifting property of 2nd order random variable

$$\omega_i^x = \omega_i^e$$

$$\mu_i^x = \mu_i^e + p_n$$

$$s_i^x = s_i^e$$

Difference between speech and excitation's mixture parameters are only mean parameters

Linear prediction Mixture density network

LP-STRUCTURED MDN

LP-MDN-based neural vocoding

1. Mixture parameter prediction

$$\left[\mathbf{z}_n^{\omega}, \mathbf{z}_n^{\mu}, \mathbf{z}_n^{s}\right] = NeuralVocoder\left(\mathbf{x}_{<n}, \mathbf{h}_n\right)$$

2. Compute prediction term

$$p_n = \sum_{i=1}^{P} \alpha_{n,i} x_{n-i}$$

3. Mixture parameter modification

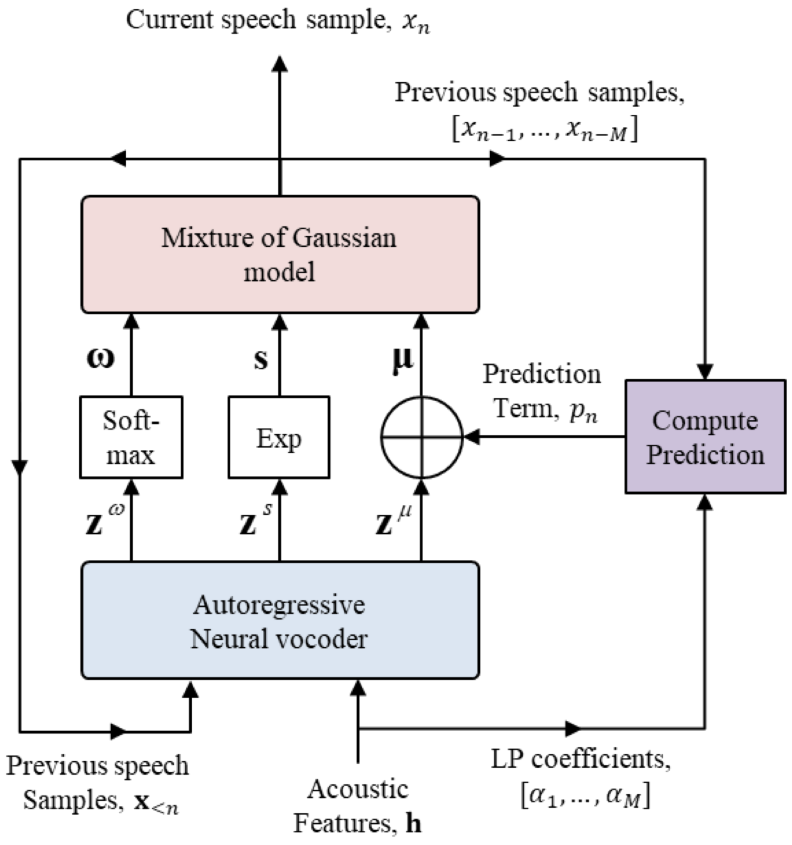
$$\boldsymbol{\omega}_n = \mathrm{softmax}(\mathbf{z}_n^{\omega})$$
$$\boldsymbol{\mu}_n = \mathbf{z}_n^{\mu} + p_n$$
$$\mathbf{s}_n = \exp(\mathbf{z}_n^{s})$$

4. MoG likelihood calculation

$$p(x_n \mid \mathbf{x}_{<n}, \mathbf{h}_n) = \sum_{i=1}^{N} \omega_{n,i} \cdot \frac{1}{\sqrt{2\pi} s_{n,i}} \exp\left[-\frac{(x_n - \mu_{n,i})^2}{2 s_{n,i}^2}\right]$$

5. Train the network to minimize negative log-likelihood loss

$$\mathrm{L}_{nll} = \sum_{n} [-\log p(x_n \mid x_{<n}, \mathbf{h}_n)]$$



[Neural vocoder with LP-MDN framework]

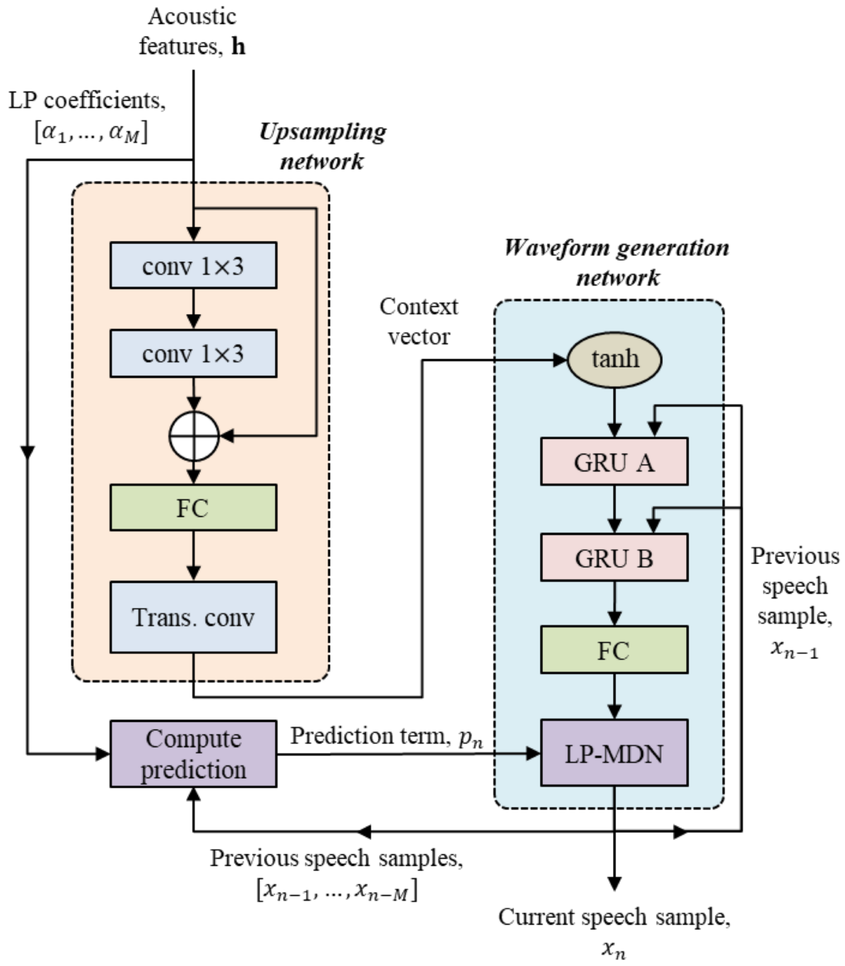
iLPCNet vocoder

Upsampling network

- Match the time-resolution of acoustic features to the sampling rate of speech signal
- Architecture
  - Two stacks of convolution layer
    - Extract contextual information of acoustic features
  - Transposed convolution layer
    - Upsample the context features

Waveform generation network

- Autoregressively generate waveform samples
- Architecture
  - Two stacks of gated recurrent unit (GRU) layers
  - Apply LP-MDN to generate the speech's distribution



[Block diagram of iLPCNet vocoder]

Effective training and generation methods

Short-time Fourier transform (STFT)-based power loss

$$\mathrm{L}_{pl} = \left\| STFT(\mathbf{x}) - STFT(\hat{\mathbf{x}}) \right\|_2^2$$

$$\Rightarrow \mathrm{L} = \mathrm{L}_{nll} + \lambda \mathrm{L}_{pl}$$

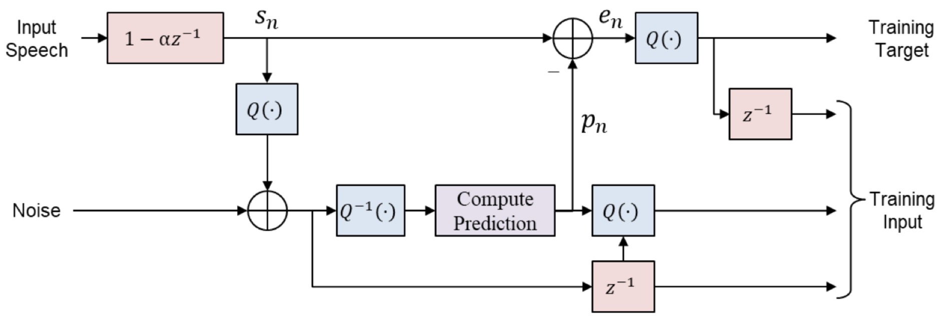
- Capture the time-frequency distribution of the speech waveform

Continuous training noise injection

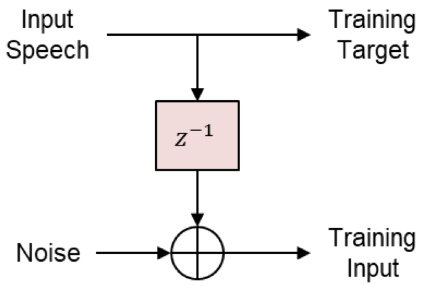
$$\hat{x}_{n-1} = x_{n-1} + \frac{4}{2^{16}}\varepsilon, \text{ where } \varepsilon \sim \mathrm{N}(0,1)$$

$$\Rightarrow x_n = iLPCNet(\hat{x}_{n-1}, \mathbf{h}_n)$$

- Train the propagated prediction error via autoregressive connection
- Simplify complicated noise injection pipeline of original LPCNet



[Noise injection process of **LPCNet**]



[Noise injection process of **iLPCNet**]

Effective training and generation methods

Conditional sampling for MoG distribution

- Conventional random sampling method

$$x_{rand} \sim \mathrm{N}\,(\mu, s)$$

➡ Noisy artifacts in the voiced region

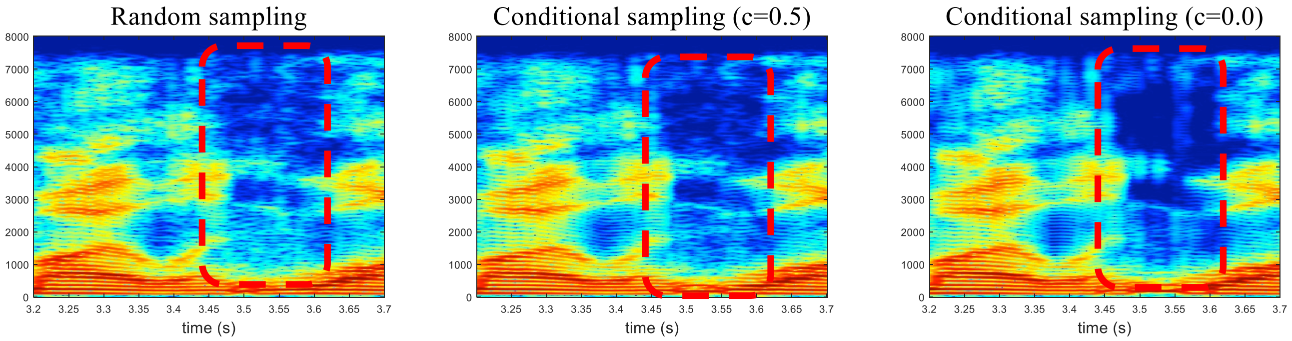
- Distribution sharpening method

$$x_{sharp} \sim \mathrm{N}\,(\mu, c \cdot s),\ \text{where } c < 1$$

➡ Eliminate noisy artifacts by reducing noise component

- Proposed conditional sampling method

$$x = \underbrace{vuv \cdot x_{sharp}}_{\text{Sharpened sampling at the voiced region}} + \underbrace{(1 - vuv) \cdot x_{rand}}_{\text{Random sampling at the unvoiced region}}$$



[Spectrogram example of conditional sampling]

COMPARISON WITH ORIGINAL LPCNET

	LPCNet	Proposed iLPCNet
Distribution type	**Discrete**	**Continuous**
Method to reflect LP structure	**Feeding LP-related signals,** $[e_{n-1}, x_{n-1}, p_n]$, into GRU	**LP-MDN**
	Open-loop solution	**Closed-loop** solution
Target of WaveRNN	**Excitation**	**Speech**
Tuning methods	Waveform embedding	STFT-based power loss
	Discrete noise injection	**Continuous** noise injection
	Conditional sharpening for **softmax** distribution	Conditional sharpening for **MoG** distribution

[Demo]

EXPERIMENT SETUP

Common settings

Database	Korean professional female
Sampling rate / Quantization bit	24kHz / 16 bits
Training / validation / test	4,976 (9.9 hours) / 280 / 140
Acoustic features	Extracted by ITFTE vocoder [1]
	79-dim.
	5-ms (=120 samples) frame shift
	Zero mean & unit variance normalization

Neural vocoders

- WaveNet [2]
- LPCNet [3]
- Proposed iLPCNet

Scenarios

- Analysis / synthesis (A/S) scenario
- Text-to-speech (TTS) scenario
  - Tacotron 2 acoustic model [4]

Performance evaluation

- Mean opinion score (MOS) listening test
- A-B preference test

[1] E. Song et.al., "Effective spectral and excitation modeling techniques for LSTM-RNN-based speech synthesis systems," in *IEEE/ACM Trans. ASLP*, 2017
[2] A. van den Oord et. al., "WaveNet: A generative model for raw audio," *arXiv preprint*, 2016
[3] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019.
[4] J. Shen et. al., "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram prediction," in *Proc. ICASSP*, 2018

[Demo]

EXPERIMENT SETUP

Neural vocoders

- WaveNet vocoder

Dilation	3 * [1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
Layer	30
Receptive field	3,071
Skip channels	128
Residual channels	128

- LPCNet vocoder

FC layer dimension	64
GRU A dimension	256
GRU B dimension	16
Waveform embedding dimension	256

- Proposed iLPCNet vocoder

FC layer dimension	256
Transposed convolution kernel size	120 (5-ms)
GRU A dimension	256
GRU B dimension	16
Speech distribution	Single Gaussian distribution
Power loss weight, λ	10.0
Sharpening factor, c	0.7

  - Same GRU size with LPCNet vocoder

[Demo]

EXPERIMENT SETUP

Tacotron 2 acoustic model for TTS scenario

Encoder	Character embedding	Dimension	512
	Convolution layer	Number of layers	3
		Kernel size	10×1
		Channels	512
	BiLSTM layer	Units	512
Attention	Location-sensitive attention	Dimension	128
		Kernel size	64×1
Decoder	Pre-net FC layer	Number of layers	2
		Dimension	256
	LSTM layer	Number of layers	2
		Units	1,024
	Post-net convolution layer	Number of layers	5
		Kernel size	5×1
		Channels	512

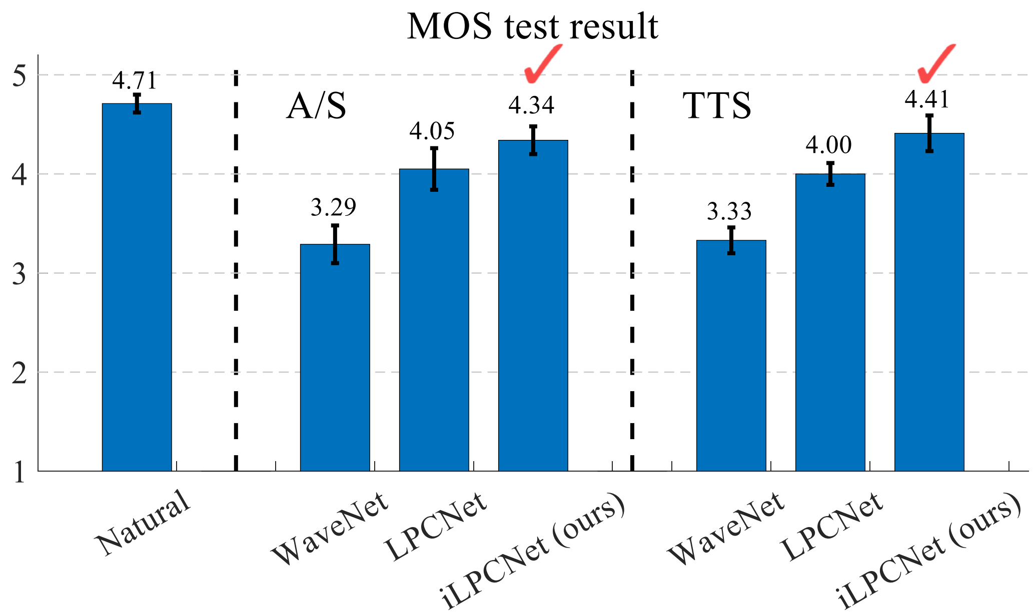
[Demo]

Performance evaluations

MOS test

- Score the quality of speech
- 15 native Korean listeners
- 15 randomly selected synthesized utterances from test set

Results



[Scoring criteria for MOS test]

Score	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

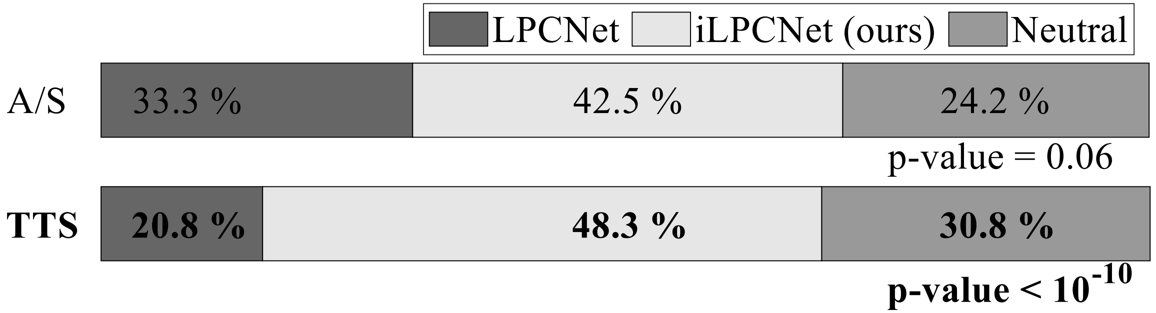
[Demo]

Performance evaluations

A-B preference test

- Rate the quality preference
- 15 native Korean listeners
- 15 randomly selected synthesized utterances from test set

Results



SUMMARY & CONCLUSION

Summary

- Proposed an improved LPCNet (iLPCNet) vocoder-based parametric TTS system

Linear prediction (LP)-structured mixture density network (MDN)

- Structurally constructed the LP structure within an autoregressive neural vocoder framework

Improved LPCNet vocoder

- Incorporated LP-MDN into LPCNet vocoder with additional effective training and generation methods
- Achieved simpler and more compact architecture by removing extra modules in LPCNet, which was designed for handling the quantization effect caused by μ-law method

Performance evaluation results

- Outperformed the conventional neural vocoding systems
  - **4.41 MOS result**
  - **27.5% higher quality preference** than conventional LPCNet vocoder

Thank you!