Voice Synthesis and Applications

- Focusing on waveform generation methods

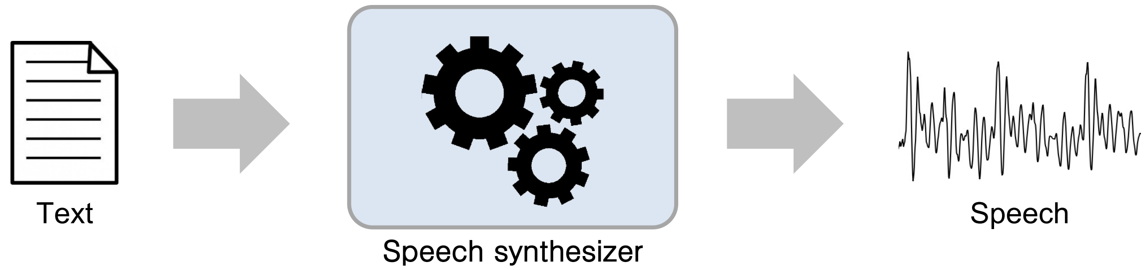
Naver Clova

Min-Jae Hwang

INTRODUCTION

Text-to-Speech (TTS) technology

Concept

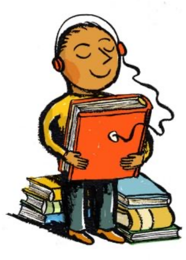


- The system synthesizing speech waveform from given input text

Application area

Navigation

AI speaker



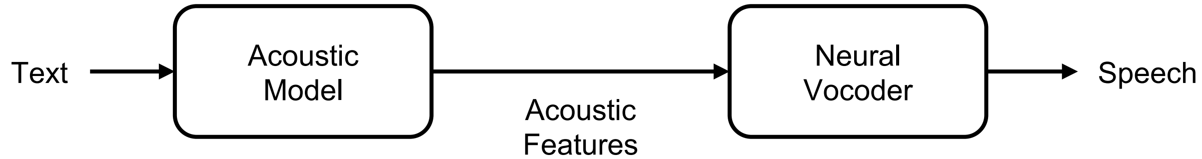
Audiobook

Ai Call

Speech translation

GENERAL ARCHITECTURE OF TTS SYSTEM

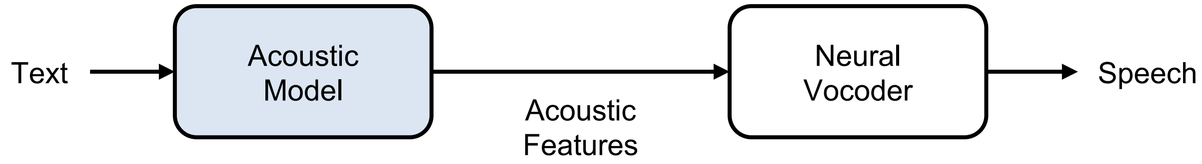
Overview



[End-to-end TTS system]

GENERAL ARCHITECTURE OF TTS SYSTEM

Overview

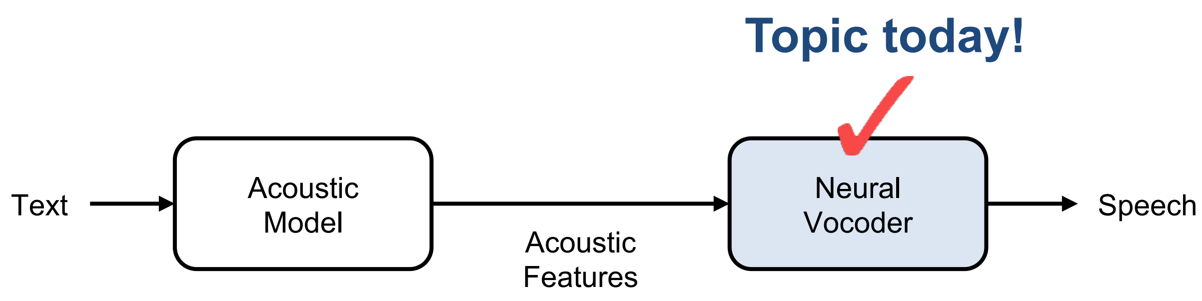


[End-to-end TTS system]

- Acoustic model
  - Generate speech's acoustic feature from input text
  - Acoustic features?
    - Mel-spectrogram, pitch, energy, or spectral envelope, etc.
  - Famous models
    - Tacotron [1] and FastSpeech [2]

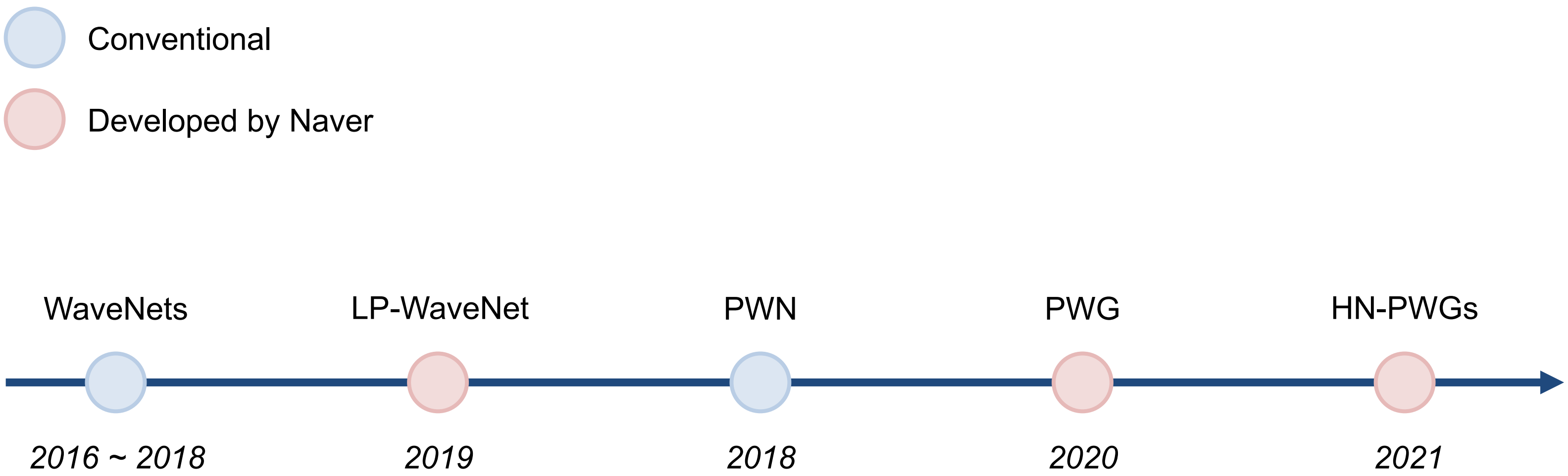
General architecture of TTS system

Overview



[End-to-end TTS system]

- Neural vocoder
  - Synthesize speech waveform from generated acoustic features
  - Famous models
    - WaveNet [3] and Parallel WaveGAN [4]



Neural vocoder

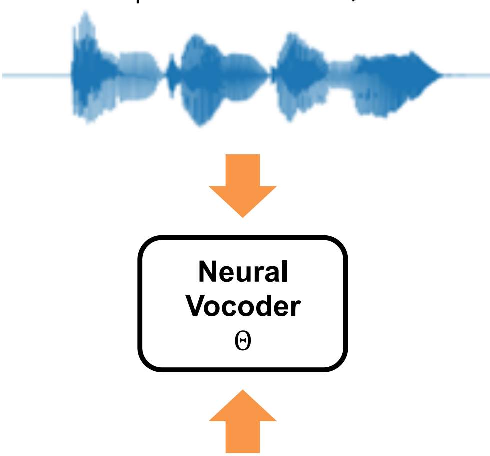
Overview

Neural vocoder

[Training phase]

$$\hat{\Theta} = \arg\max_{\Theta} p(\mathbf{x} \mid \mathbf{h}, \Theta)$$

Speech waveform, $\mathbf{x}$



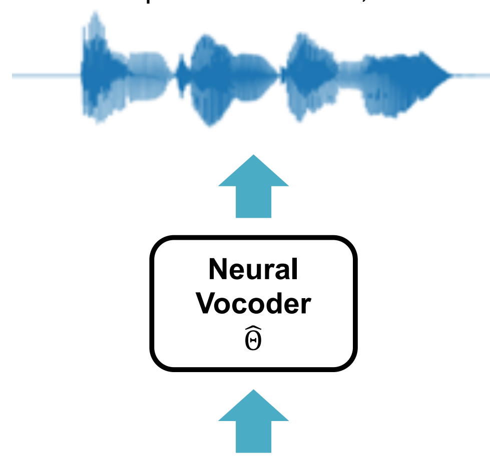
Acoustic
Feature, $\mathbf{h}$

Optimize network parameters
to maximize the likelihood of speech waveform

[Inference phase]

$$\hat{\mathbf{x}} \sim p(\mathbf{x} \mid \mathbf{h}, \hat{\Theta})$$

Speech waveform, $\hat{\mathbf{x}}$



Acoustic
Feature, $\mathbf{h}$

Sample speech waveform from
estimated speech likelihood

Neural vocoder

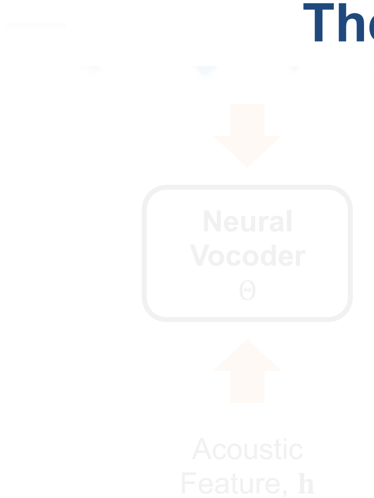
[Training phase]

$$\hat{\Theta} = \arg\max_{\Theta} p(\mathbf{x} \mid \mathbf{h}, \Theta)$$

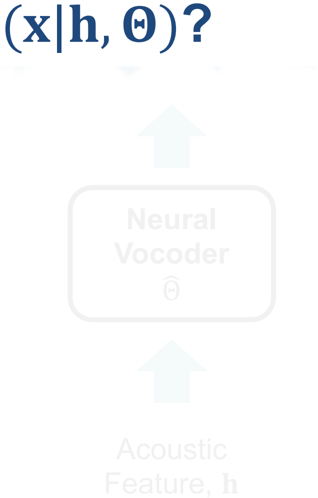
[Inference phase]

$$\hat{\mathbf{x}} \sim p(\mathbf{x} \mid \mathbf{h}, \hat{\Theta})$$

Generative model is essential!
Then, how does it define $p(\mathbf{x}|\mathbf{h}, \boldsymbol{\Theta})$?



Optimize network parameters to maximize the likelihood of speech waveform



Sample speech waveform from estimated speech likelihood

Neural vocoder

[Training phase]

$$\hat{\Theta} = \arg\max_{\Theta} p(\mathbf{x} \mid \mathbf{h}, \Theta)$$

[Inference phase]

$$\hat{\mathbf{x}} \sim p(\mathbf{x} \mid \mathbf{h}, \hat{\Theta})$$

Generative model is essential!
Then, how does it define $p(\mathbf{x}|\mathbf{h}, \Theta)$?

Neural Vocoder Θ

Neural Vocoder $\hat{\Theta}$

Autoregressive (AR) Approach

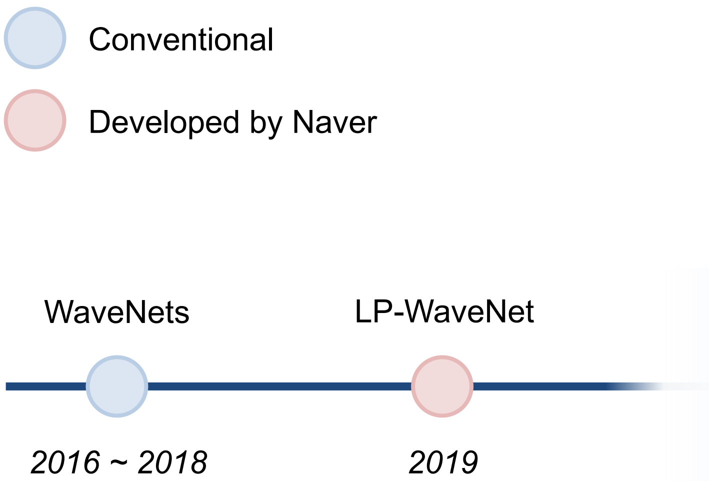
Non-AR Approach

Feature, $\mathbf{h}$

Feature, $\mathbf{h}$

Optimize network parameters to maximize the likelihood of speech waveform

Sample speech waveform from estimated speech likelihood



Neural vocoder

Autoregressive models

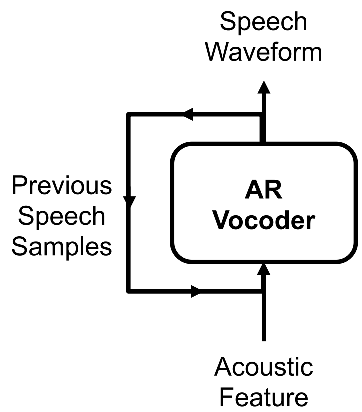
AR NEURAL VOCODER

Probability model

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{n=0}^{T-1} p(x_n \mid \mathbf{x}_{<n}, \mathbf{h})$$

Neural vocoder's target

- Factorize speech's probability as a product of conditional probabilities for given past speech samples



[Concept of AR vocoder]

Inputs

- (1) Acoustic features
- (2) Previously generated samples

Output

$$p(x_n|\mathbf{x}_{<n}, \mathbf{h}) = NeuralVocoder(\mathbf{x}_{<n}, \mathbf{h})$$

- Probability of current speech sample

WaveNet vocoder

First AR generative model for raw waveform [3]

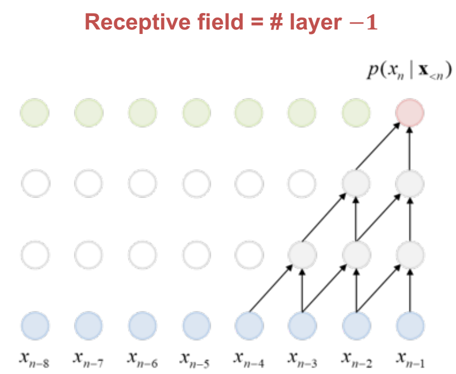
Key feature

- Multiply stacked ***dilated causal convolution*** layers
  - Exponentially increase the receptive field
  - Effectively capture speech's long-term correlation problem

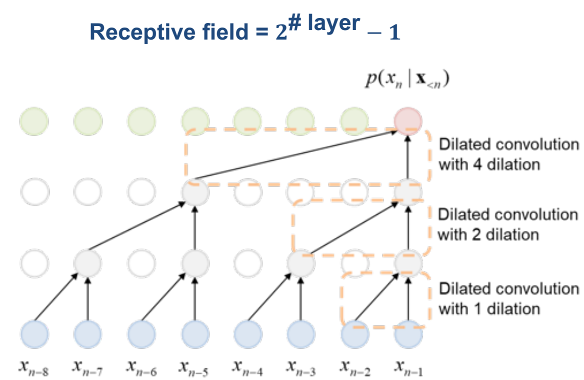
Various types of WaveNet vocoder

- μ-law WaveNet [5]
- Mixture density network (MDN)-based WaveNet [1, 10]

→ ***Depending on how to define the speech distribution***



[WaveNet with standard causal convolution]



[WaveNet with dilated causal convolution]

Various WaveNet Vocoders

μ-law WaveNet [5]

- Re-define speech distribution as ***discretized symbols***
  - (1) Apply μ-law companding to obtain evenly distributed speech signal

$$y = sign(x) \cdot \frac{\ln\left(1+\mu|x|\right)}{\ln(1+\mu)}, \ \mu{=}255$$

  - (2) Apply 8-bit one-hot encoding

$$p = OneHot_{8bit}(y)$$

    - Discretize speech sample in ***256 symbols***

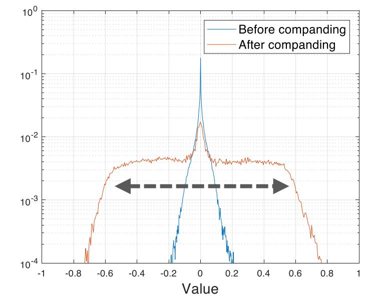
- Use WaveNet to solve multi-class ***classification problem***
  - Predict discretized speech symbols

$$\mathbf{z}^q = WaveNet(\mathbf{q}_{<n} \mid \mathbf{h}) \Rightarrow q_n = \frac{\exp(z_n^q)}{\sum_i \exp(z_i^q)}$$

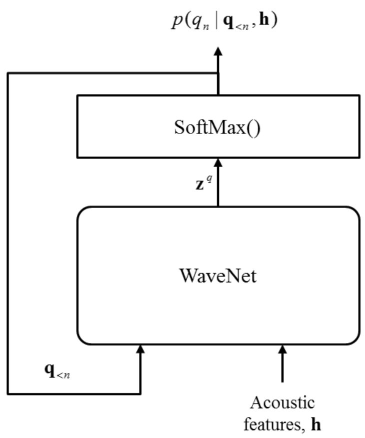
  - Optimize to minimize cross-entropy (CE) loss

$$L = \sum_n \left[-p_n \log q_n\right]$$

- Advantages
  - Provide better quality than conventional rule-based vocoders
    - Free from rule-based vocoder's heuristic signal processing pipeline



[Distribution of speech samples]

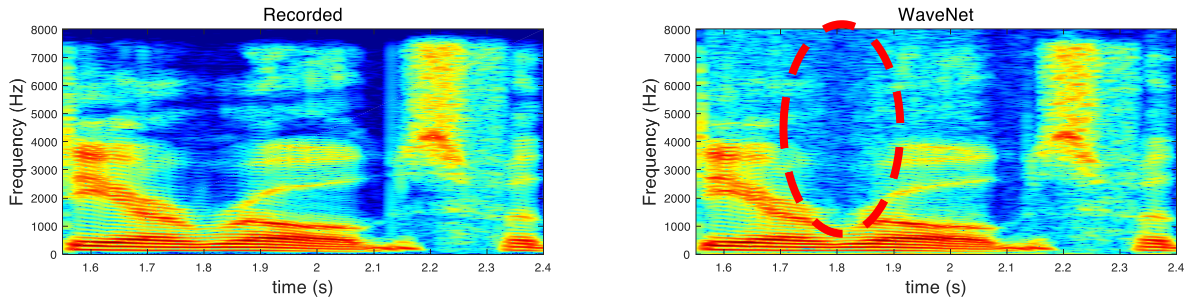


[μ-law WaveNet]

VARIOUS WAVENET VOCODERS

Limitation of μ-law WaveNet

- Noisy synthetic speech due to rough quantization of waveform



Naive solution

- Consider that waveform is usually discretized by 16-bits quantization method
  → Expand the softmax dimension to 65,536 (=2^{16})

  Expensive computational cost & difficult to train

Mixture density network (MDN)-based solution [1, 10]

- Train the WaveNet to predict the parameter of pre-defined speech distribution

VARIOUS WAVENET VOCODERS

MDN-WaveNet

- Define the speech distribution as ***mixture of Gaussian (MoG) distribution***

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

- Use WaveNet for MDN modeling [6]
  - Predict ***mixture parameters***

    $$[\mathbf{z}^{\pi}, \mathbf{z}^{\mu}, \mathbf{z}^{s}] = WaveNet(\mathbf{x}_{<n}, \mathbf{h})$$

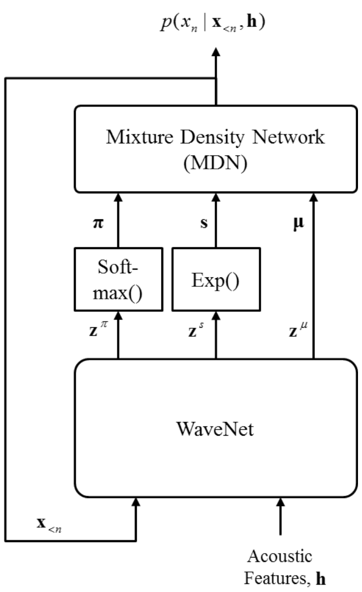
    $\boldsymbol{\pi} = \mathrm{softmax}(\mathbf{z}^{\pi})$, for unity-summed mixture gain

    $\boldsymbol{\mu} = \mathbf{z}^{\mu}$

    $\mathbf{s} = \exp(\mathbf{z}^{s})$, for positive value of mixture scale

  - Optimize network by negative log-likelihood (NLL) loss

    $$L = \sum_n [-\log p(x_n \mid x_{<n}, \mathbf{h})]$$



[MDN-WaveNet]

Mixture density network

MDN-WAVENET

Advantage

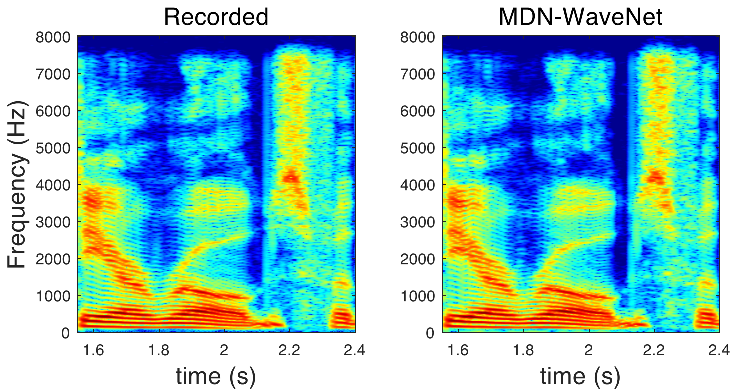
- Enable to model the continuously distributed speech waveform
  → Provide higher quality than μ-law WaveNet

Problem

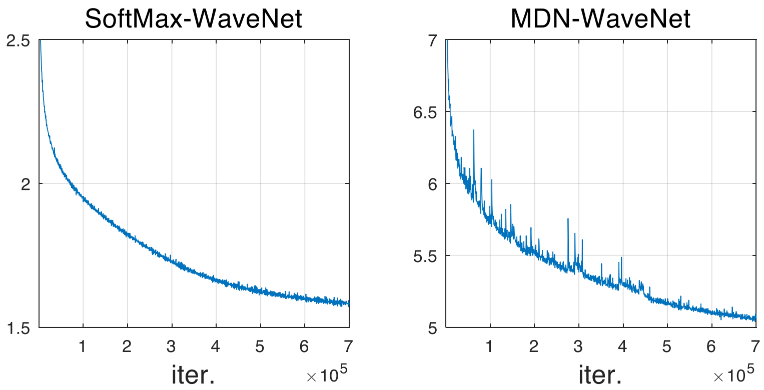
- ***Difficult to train*** due to increased target distribution's degree of freedom

Solution based on the human's speech production model [7]

- Model **the vocal source signal**, whose physical behavior is much simpler than the speech signal



[Spectrogram comparison]

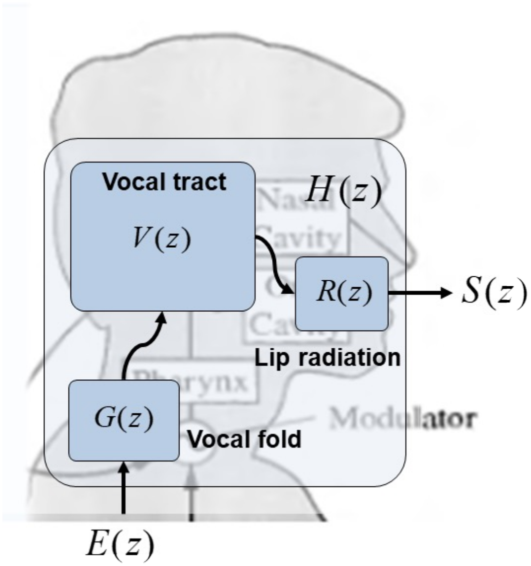


[Loss comparison]

SPEECH PRODUCTION MODEL

Source-filter theory of speech production [7]

- Modeling the speech as the filtered output of ***vocal source*** signal to ***vocal tract filter***



[Speech production model]

$$S(z) = \left[G(z) \cdot V(z) \cdot R(z)\right] \cdot E(z)$$

Speech = [vocal fold × vocal tract × lip radiation] × excitation

SPEECH PRODUCTION MODEL

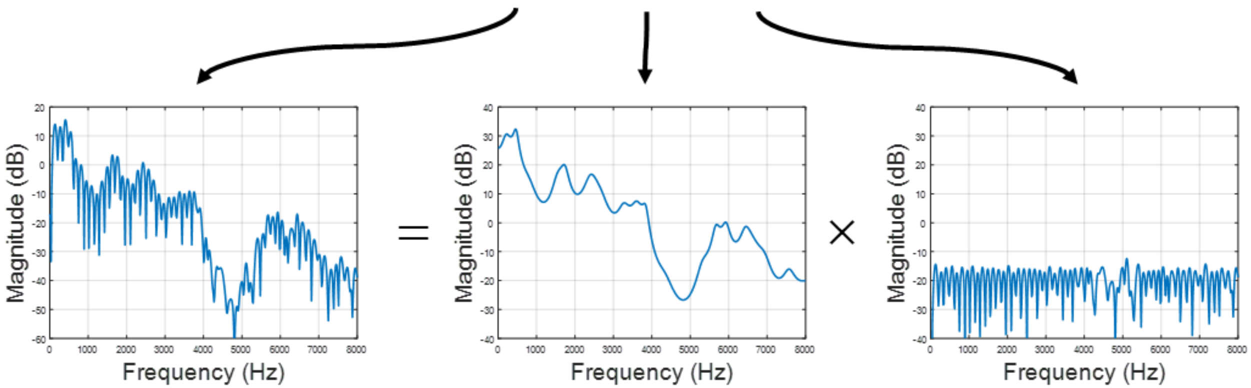
Decouple vocal source & tract by using *linear prediction (LP) analysis* [7]

- Define speech signal as linear combination of past speech samples

$$s_n = \sum_{i=1}^{P} \alpha_i s_{n-i} + e_n \quad \Leftrightarrow \quad S(z) = H(z) \cdot E(z), \text{ where } H(z) = \frac{1}{1 - \sum_{i=1}^{p} \alpha_i z^{-i}}$$

Vocal tract part = LP coefficients, $(= \alpha_i)$
Vocal source part = Error signal of LP analysis, $(= e_n)$

$$S(z) = H(z) \cdot E(z)$$



[Spectral deconvolution through LP analysis]

Linear prediction   Mixture density network

LP-STRUCTURED MDN

Mathematical assumption for AR vocoder

- Consideration about linear prediction term, p_n
  1. Previous speech samples, $\mathbf{x}_{<n}$, are given
  2. LP coefficients, $\{\alpha_i\}$, indicating spectral envelope of speech, are given

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

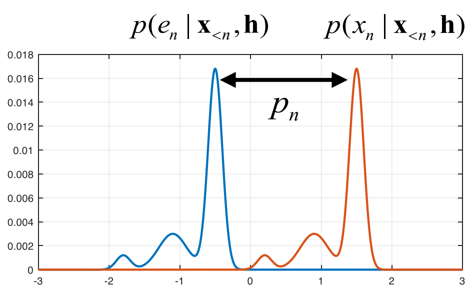
- Random variables (RVs) of speech X_n and excitation E_n

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

  X_n and E_n have only ***constant difference*** *of p_n*

- Parametrize RVs by using mean and variance

  Difference between X_n and E_n is ***only mean parameter***



[Probabilistic relationship between speech and excitation]

Linear prediction Mixture density network

LP-STRUCTURED MDN

LP-MDN

- Formulate the relationship between speech and excitation within MDN approach [8]

- (1) Predict MoG parameters of excitation signal by using neural vocoder

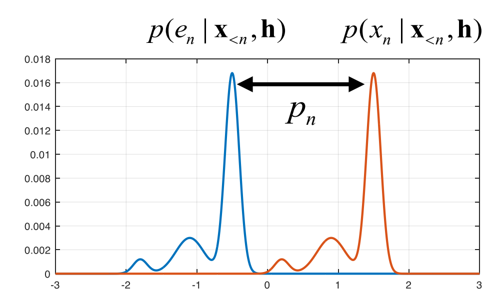
$$p(e_n|\mathbf{x}_{<n}, \mathbf{h}_n) \sim \sum_n \omega_i^e \cdot N(\mu_i^e, s_i^e)$$

- (2) Shift only mean parameters by p_n

$$\omega_i^x = \omega_i^e$$
$$\boxed{\mu_i^x = \mu_i^e + p_n}$$
$$s_i^x = s_i^e$$

- (3) Compute likelihood of speech signal

$$p(x_n|\mathbf{x}_{<n}, \mathbf{h}_n) \sim \sum_n \omega_i^x \cdot N(\mu_i^x, s_i^x)$$



[Probabilistic relationship between speech and excitation]

LP-WaveNet vocoder

LP-WaveNet = MDN-WaveNet + LP-MDN [8]

1. Mixture parameter prediction

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

2. Compute linear prediction term

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

3. Mixture parameter modification

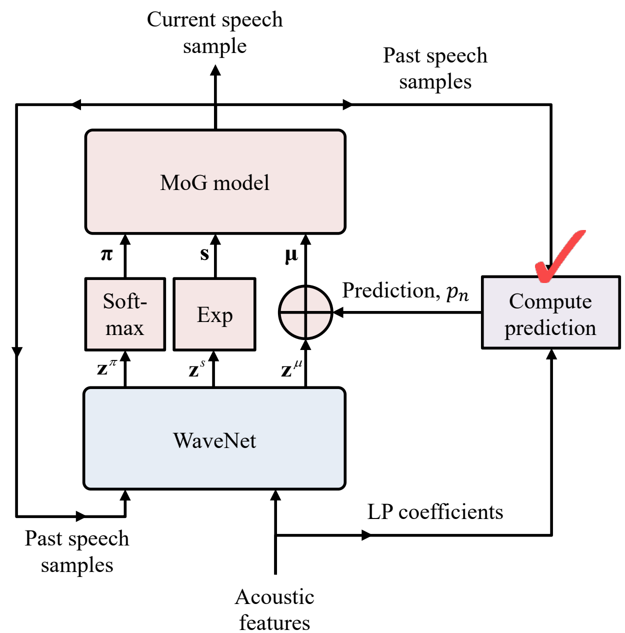
$$\boldsymbol{\omega}_n = \mathrm{softmax}(\mathbf{z}_n^{\omega})$$
$$\boldsymbol{\mu}_n = \mathbf{z}_n^{\mu} + p_n$$
$$\mathbf{s}_n = \exp(\mathbf{z}_n^{s})$$

4. MoG likelihood calculation

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

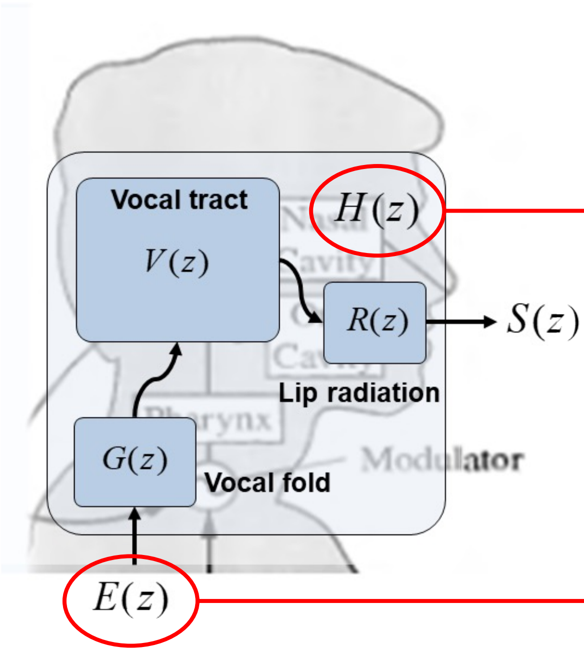
5. Train the network to minimize NLL loss

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



[LP-WaveNet]

LP-WaveNet vocoder



[Speech production model]

Modeled by LP-MDN

+

Only need to be trained

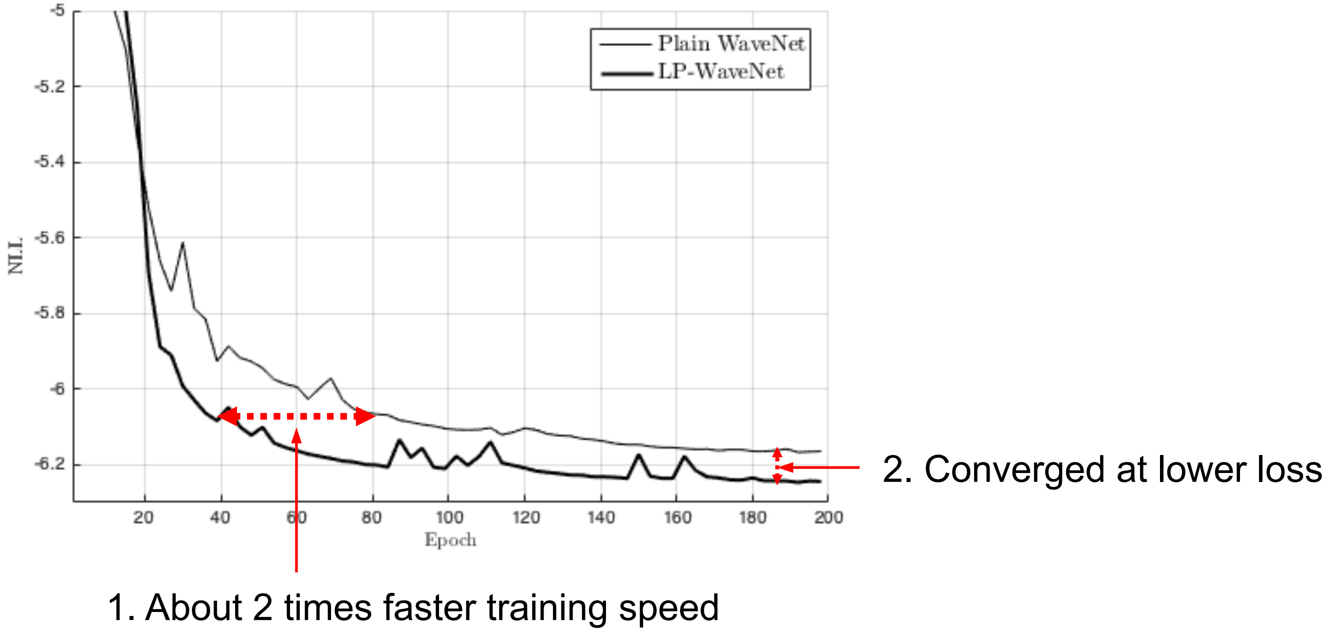
=

Training efficiency will be improved!

LP-WaveNet vocoder

Training efficiency

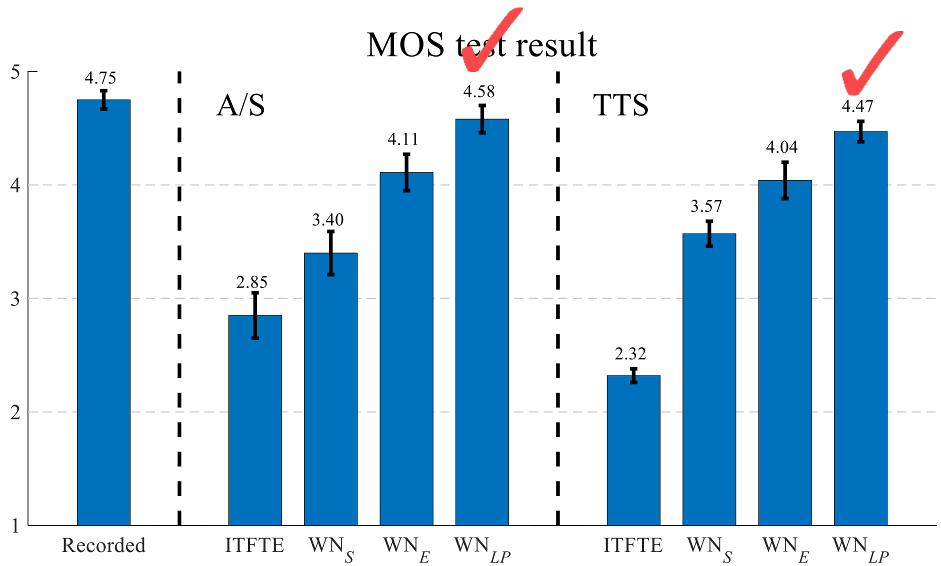
- Comparing to MDN-WaveNet



LP-WaveNet vocoder

Subjective evaluation results

- Mean opinion score (MOS) test



Provided significantly higher quality than conventional vocoders

ITFTE: Baseline rule-based vocoder [10]
WN_S: μ-law WaveNet estimating speech signal
WN_E: μ-law WaveNet estimating excitation signal
WN_{LP}: LP-WaveNet

A/S: analysis / synthesis

[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

LP-WaveNet vocoder

Industrial contribution to Naver's various TTS services

Navigation

AI speaker

Ai Call

News reading

Limitation

- ***Very slow inference speed*** due to AR generation process
  - e.g., 300 real-time factor (RTF) even in V100 GPU environment
- ***Unsuitable for real-time TTS service***
  - e.g., Audiobook synthesis or controllable TTS, etc

k RTF: k sec. is required to synthesize 1 sec. of speech

LP-WaveNet vocoder

Industrial contribution to Naver's various TTS services

Navigation

AI speaker

Ai Call

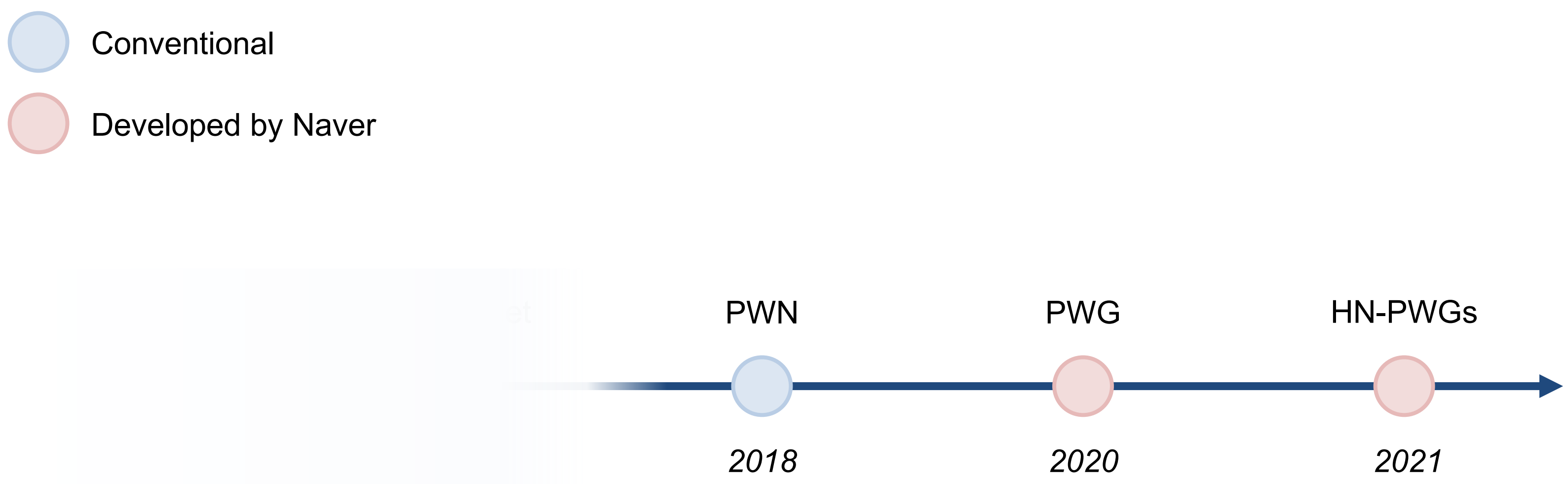
News reading

Limitation

- *Very slow inference speed* due to AR generation process
  - e.g., 300 real-time factor (RTF) even in V100 GPU environment
- *Unsuitable for real-time TTS service*
  - e.g., Audiobook synthesis or controllable TTS, etc

Developing real-time and high-quality neural vocoder has become important.

k RTF: k sec. is required to synthesize 1 sec. of speech



NEURAL VOCODER

NON-AUTOREGRESSIVE MODELS

NON-AR NEURAL VOCODER

Probability model

$$p(\mathbf{x} \mid \mathbf{h}) = \prod_{n=0}^{T-1} \underbrace{p(x_n \mid \mathbf{h})}_{\text{Neural vocoder's target}}$$

- Ignore dependency between adjacent speech samples

Inputs

- Acoustic features

Output

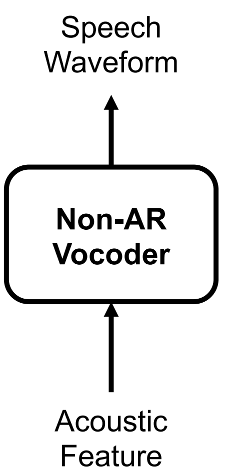
$$p(\mathbf{x}|\mathbf{h}) = NeuralVocoder(\mathbf{h})$$

- Generate entire speech samples in parallel

  ➡ Enable parallel training/generation of waveform

Limitation

- ***Worse quality than AR neural vocoder***



[Concept of non-AR vocoder]

NON-AR NEURAL VOCODER

Why non-AR model is worse than AR model?

[AR model]

$$p(x_n|\mathbf{x}_{<n}, \mathbf{h})$$

[Non-AR model]

$$p(x_n|\mathbf{h})$$

Non-AR neural vocoder

Why non-AR model is worse than AR model?

[AR model]

$$p(x_n|\mathbf{x}_{<n}, \mathbf{h})$$

Contextual information helps vocoder
to learn waveform distribution

High quality! ☺

[Non-AR model]

$$p(x_n|\mathbf{h})$$

NON-AR NEURAL VOCODER

Why non-AR model is worse than AR model?

[AR model]

$$p(x_n|\mathbf{x}_{<n}, \mathbf{h})$$

[Non-AR model]

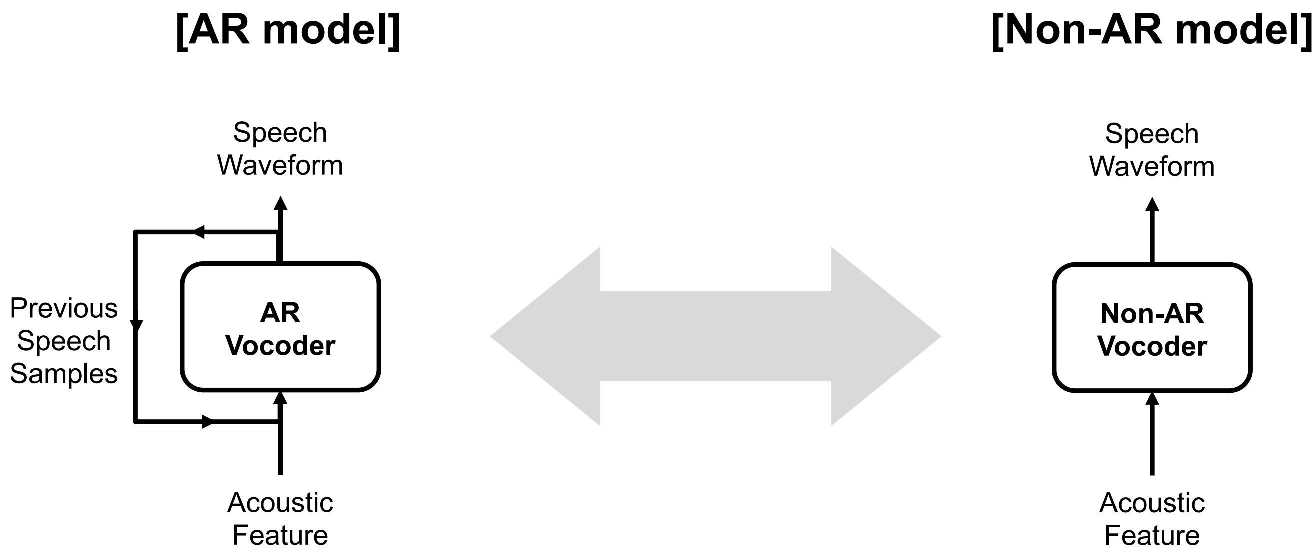
$$p(x_n|\mathbf{h})$$

Vocoder should learn speech distribution relying on only acoustic features

Unsatisfactory quality! ☹

NON-AR NEURAL VOCODER

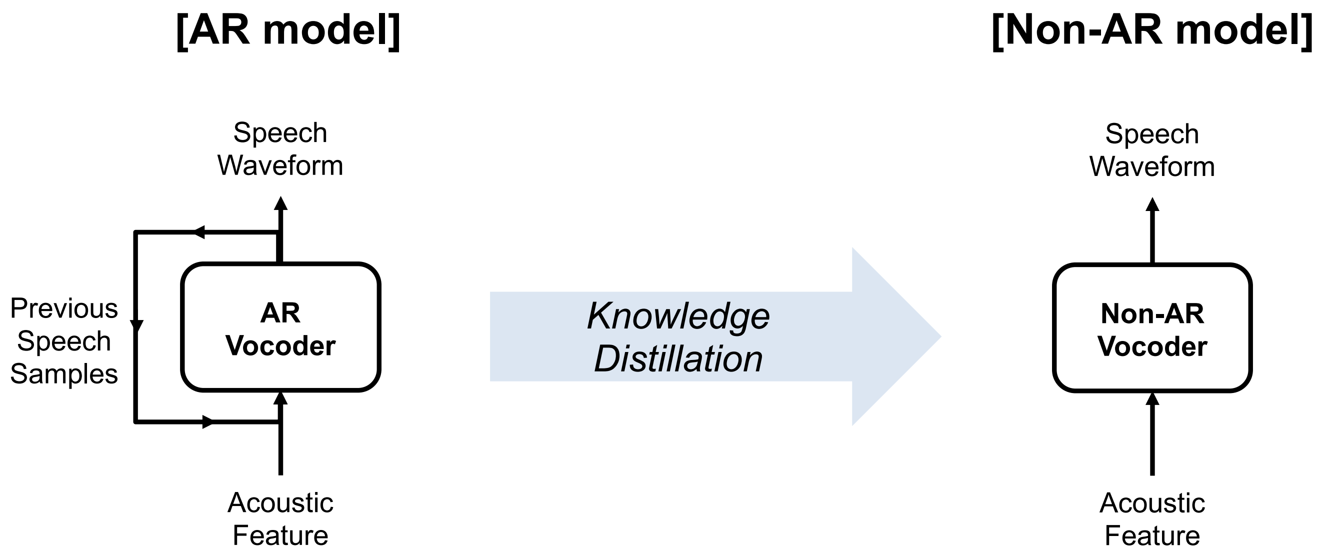
Why non-AR model is worse than AR model?



How to bridge the gap between AR and non-AR vocoders?

NON-AR NEURAL VOCODER

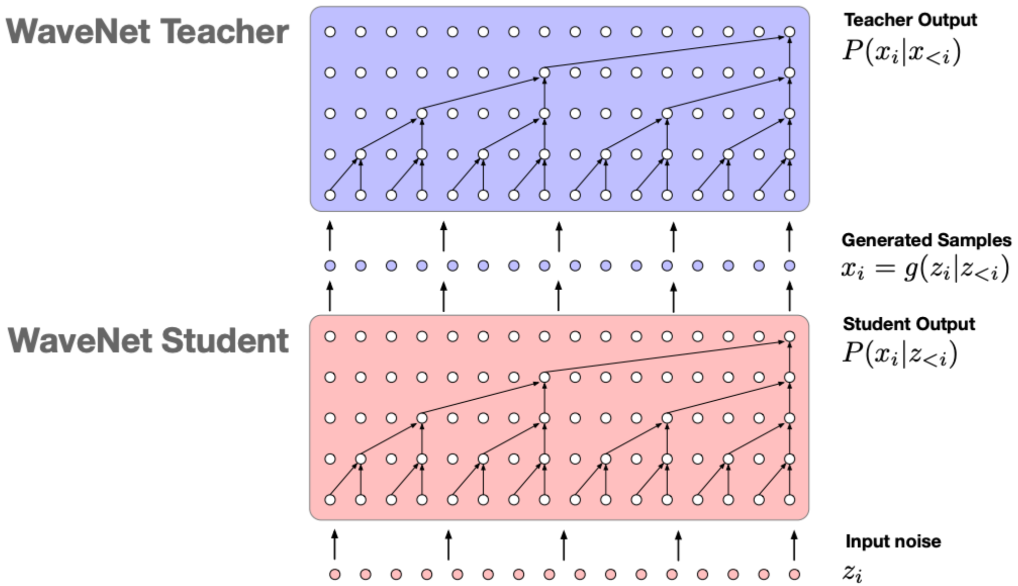
Teacher-student framework-based solution



Transfer well-trained AR vocoder's performance to non-AR vocoder

Parallel WaveNet (PWN)

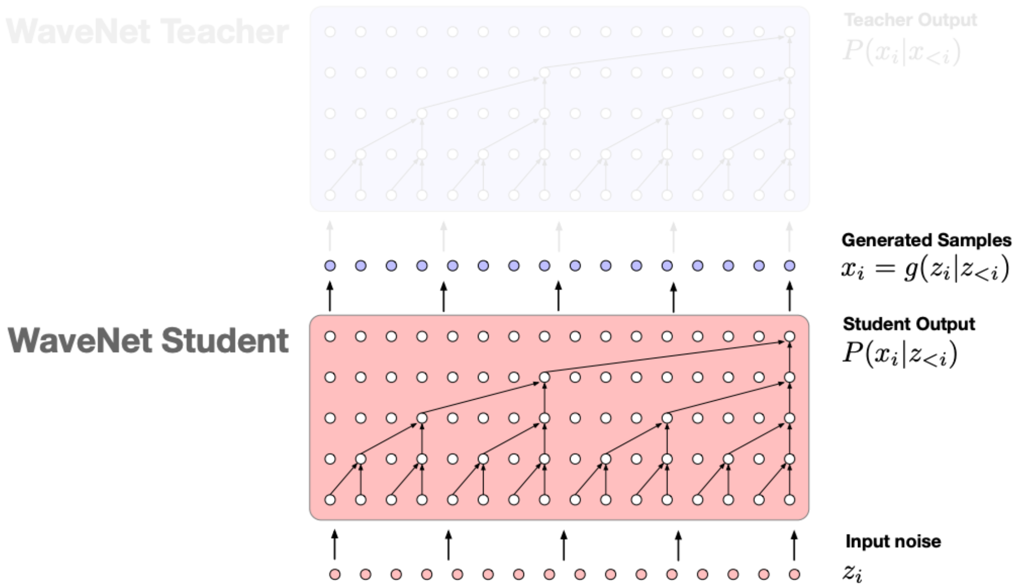
First non-AR vocoder based on teacher-student framework [10]



Guide **non-AR WaveNet** (=student) to learn
speech distribution predicted by **AR WaveNet** (=teacher)

Parallel WaveNet (PWN)

First non-AR vocoder based on teacher-student framework [10]

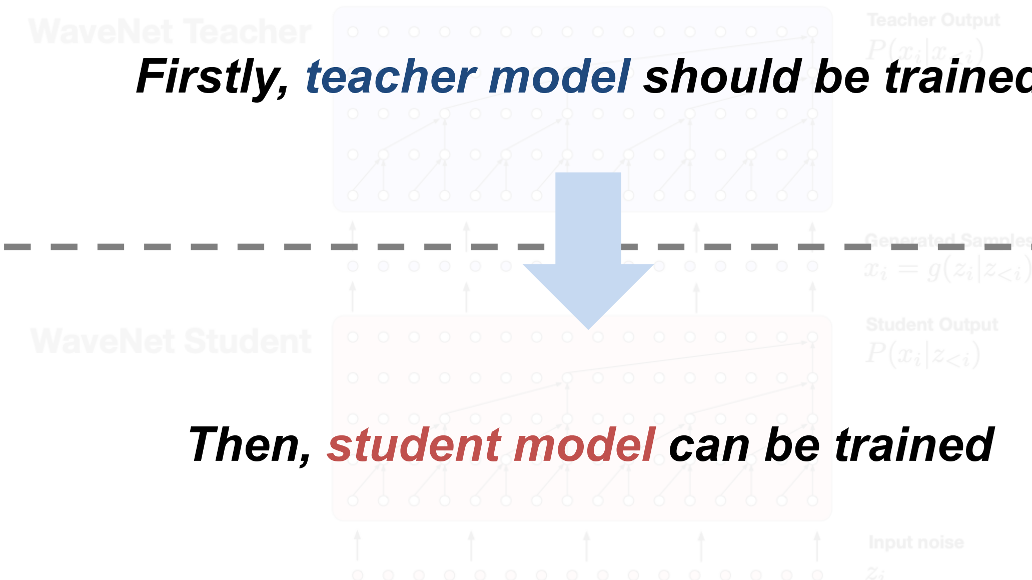


Well-distilled student WaveNet can generate **high-quality waveform** while maintaining its **fast generation speed (ex. 0.02 RTF)**

k RTF: k sec. is required to synthesize 1 sec. of speech

Parallel WaveNet (PWN)

Limitation



Firstly, teacher model should be trained

Then, student model can be trained

Two-stage training pipeline inevitably results in a long training period ☹

Ex. WaveNet (7.4 days) vs. Parallel WaveNet (12.7 days)

PARALLEL WAVEGAN (PWG)

Non-AR vocoder without teacher-student framwork [4]

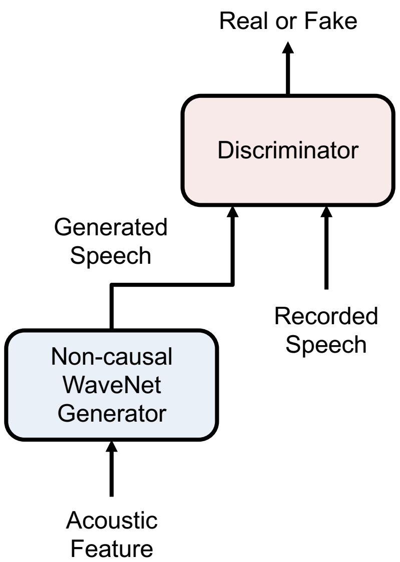
- Remove knowledge distillation process
- Instead, incorporate generative adversarial networks (GAN) framework

Key features

- (1) Non-causal WaveNet generator
  - Enable ***real-time waveform generation***
- (2) Adversarial training
  - Help the generator to produce ***realistic waveform***
- (3) Multi-resolution short-time Fourier transform (MR-STFT) loss
  - Effectively capture ***time-frequency characteristics*** of target speech

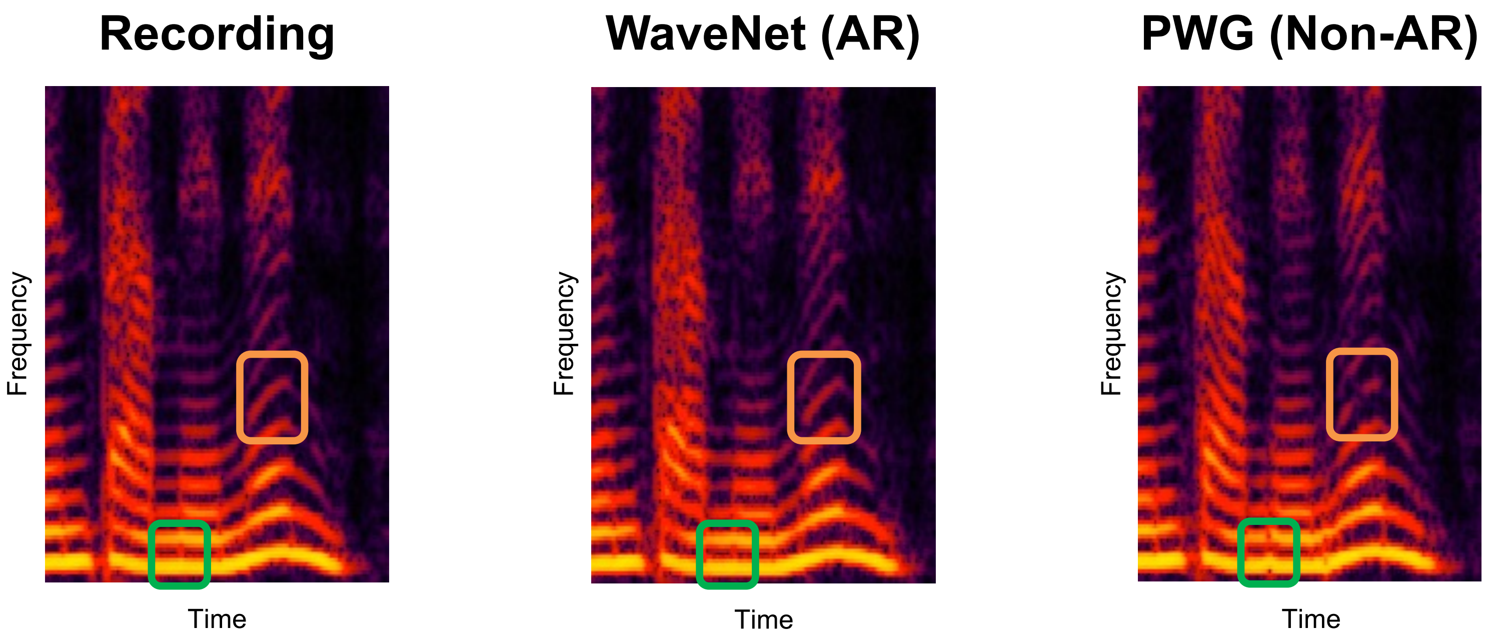
Pros and cons

- **Fast synthesis speed (e.g., 0.02 RTF)**
- **Easy to train (e.g., 3 days)**
- **Low quality of synthesized speech**



[Concept of PWG]

SPECTROGRAM EXAMPLE



Harmonic-plus-Noise Parallel WaveGAN

HN-PWG VOCODER

Adopt harmonic-plus-noise (HN) model [12] to the PWG's generator

- HN model**?**
  - **speech = *harmonic component* + *noise component***
    = Periodic, deterministic *= Aperiodic, stochastic*

Speech

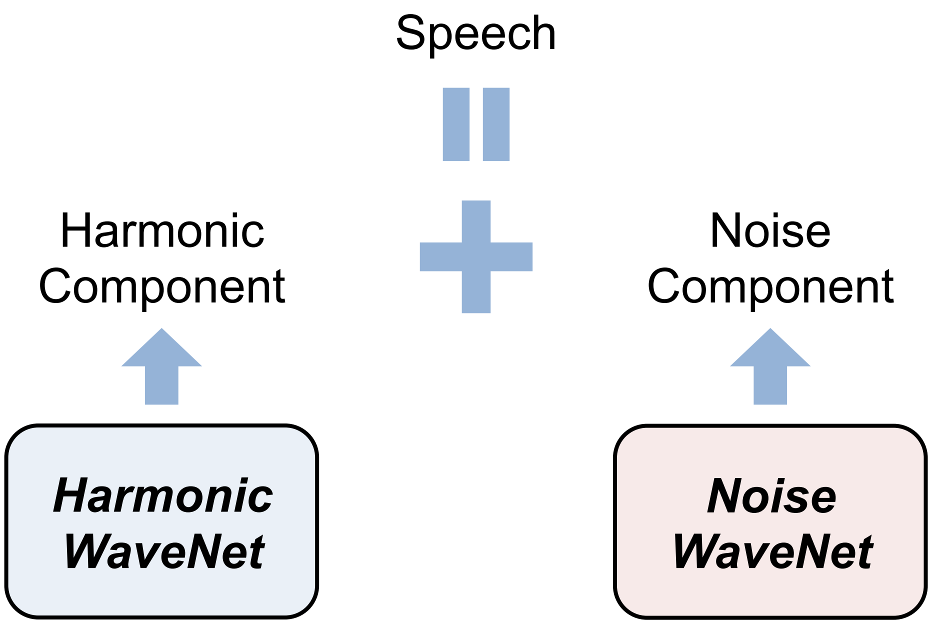
Harmonic Component

Noise Component

HN-PWG VOCODER

Adopt harmonic-plus-noise (HN) model [12] to the PWG's generator

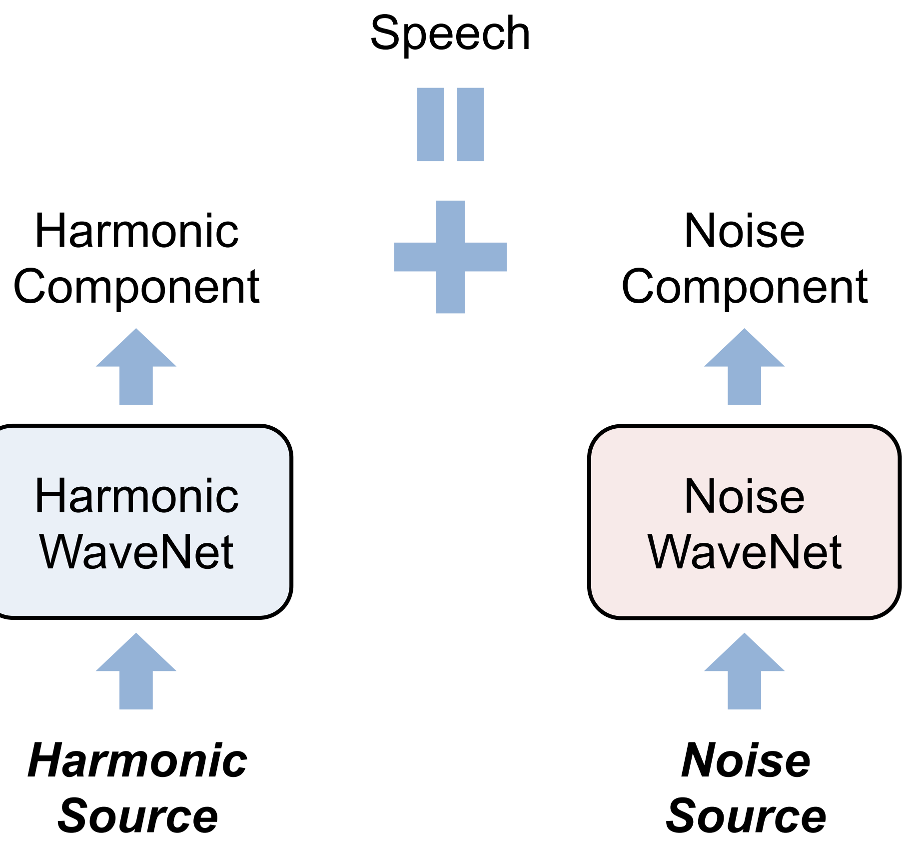
- Split WaveNet generator to two sub-WaveNet generators
  1. Harmonic WaveNet (H-WaveNet) → Generate harmonic component
  2. Noise WaveNet (N-WaveNet) → Generate noise component



HN-PWG VOCODER

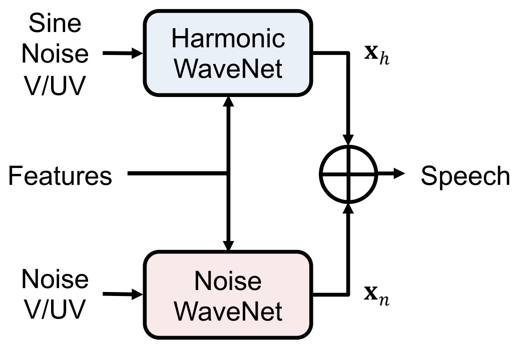
Adopt harmonic-plus-noise (HN) model [12] to the PWG's generator

- Method to impose harmonic & noise characteristics
    - Feeding harmonic- and noise-like sources to their WaveNets, respectively



HN-PWG VOCODER

Concept of HN-PWG [12]



[HN-PWG]

Source signal designs

1. Harmonic WaveNet
   - Give harmonic (=periodic) characteristic by using sinusoidal source signal

$$s[t] = \sin\left(\sum_{k=1}^{t} 2\pi \frac{f_k}{F_s} + \phi\right)$$

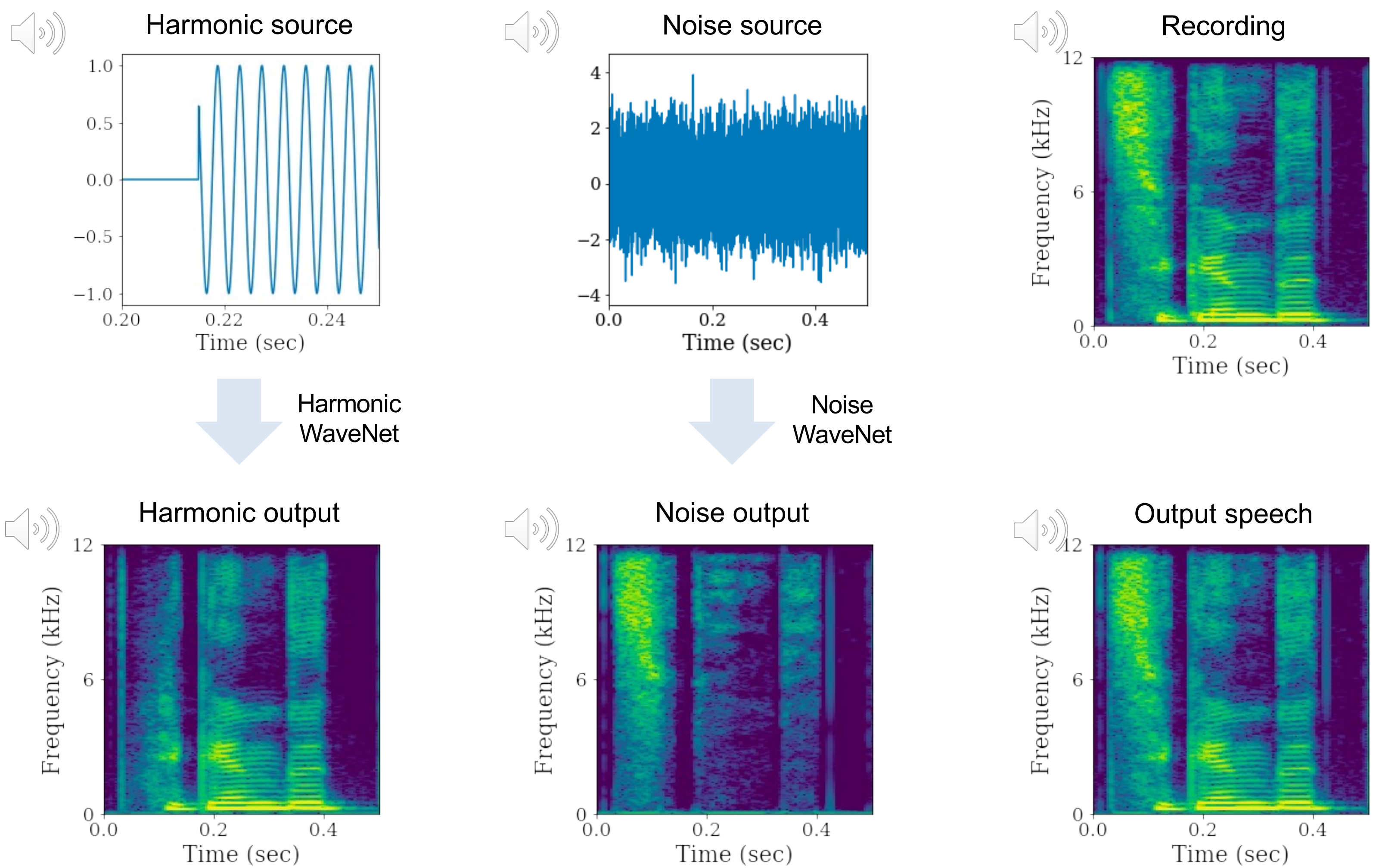
     - Design source signal to have instantaneous frequency of pitch contour

2. Noise WaveNet
   - Give noise (=aperiodic) characteristic by using Gaussian noise source signal

Harmonic-plus-Noise Parallel WaveGAN

HN-PWG VOCODER

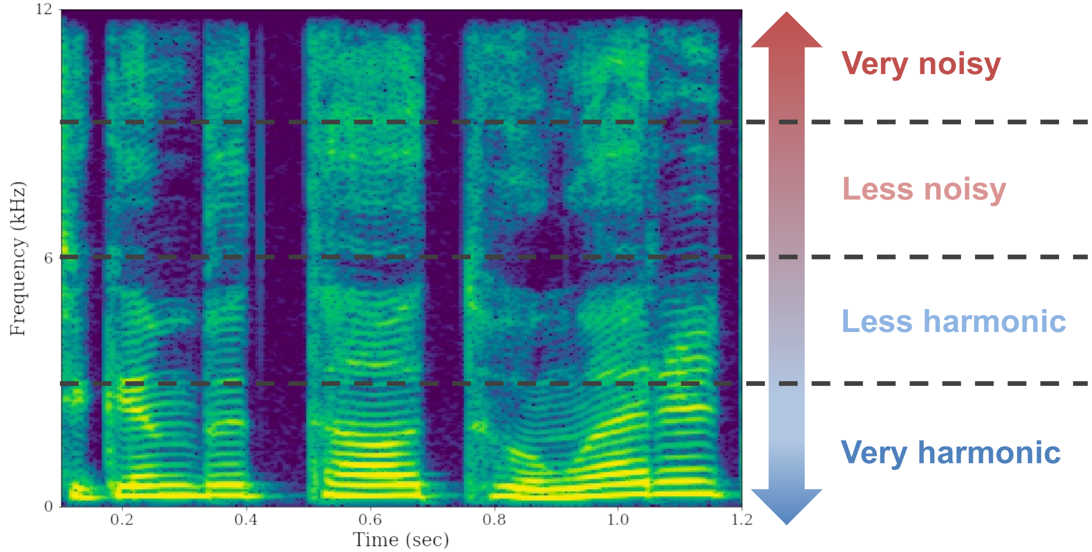
Speech sample



MULTI-BAND HN-PWG VOCODER

Consideration for the improvement of HN-PWG

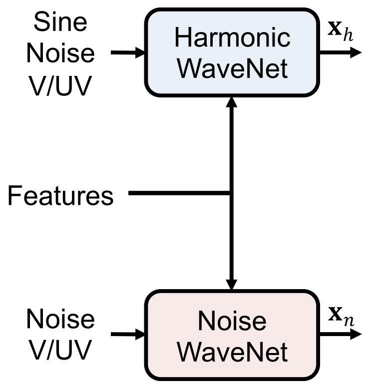
- Harmonic-noise property of speech signal
  - Low frequency band
    - Harmonic characteristic > Noise characteristic
  - High frequency band
    - Harmonic characteristic < Noise characteristic



Introduce this harmonic-noise property to the HN-PWG

MULTI-BAND HN-PWG VOCODER

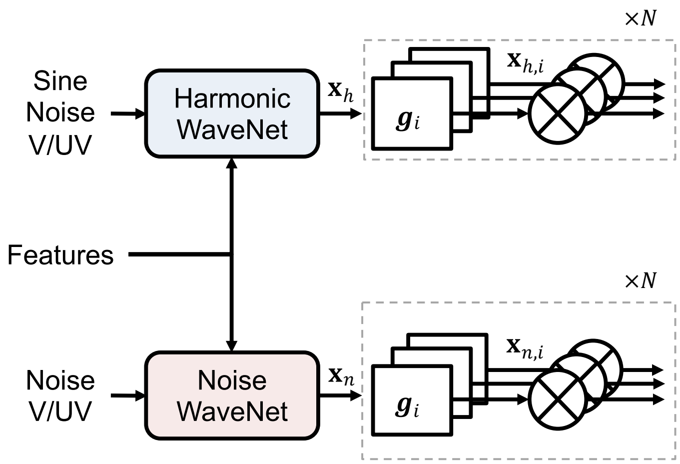
Multi-band HN-PWG [13]



Step 1.
Generate harmonic component $\mathbf{x}_h$ and noise component $\mathbf{x}_n$ by using H- and N-WaveNets

MULTI-BAND HN-PWG VOCODER

Multi-band HN-PWG [12]



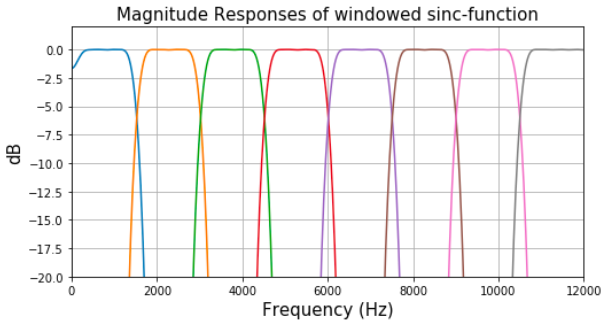
Step 2.
Decompose generated harmonic-noise components into **their subband signals** by using **windowed sinc function-based band-pass filters** (BPF; $\mathbf{g}_i$)

$$\mathbf{x}_{h,i} = \mathbf{x}_h \circledast \hat{\mathbf{g}}_i$$

$$\mathbf{x}_{n,i} = \mathbf{x}_n \circledast \hat{\mathbf{g}}_i$$

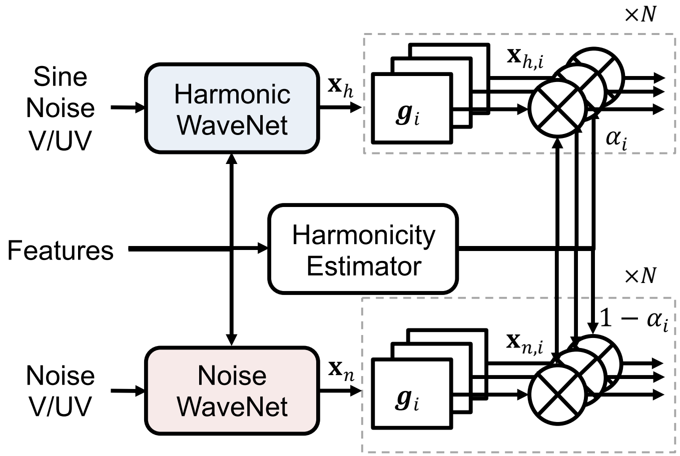
where $g_i[k] = 2f_{i+1}\mathrm{sinc}(2\pi f_{i+1}k) - 2f_i\mathrm{sinc}(2\pi f_i k),$

$$\hat{g}_i[k] = g_i[k] \cdot w_{hamm}[k]$$



MULTI-BAND HN-PWG VOCODER

Multi-band HN-PWG [12]



Step 3.
Estimate *subband harmonicity* from acoustic features

$$\{\alpha_i\} = sigmoid(CNN(\mathbf{h}))$$

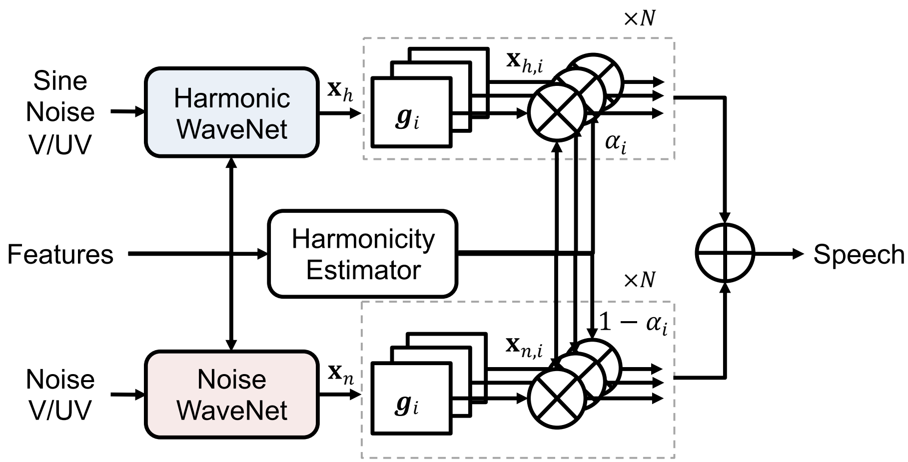
Then, adjust gain of subband signals weighted by subband harmonicity

$$\hat{\mathbf{x}}_{h,i} = \alpha_i \cdot \mathbf{x}_{h,i}$$

$$\hat{\mathbf{x}}_{n,i} = (1-\alpha_i) \cdot \mathbf{x}_{h,i}$$

MULTI-BAND HN-PWG VOCODER

Multi-band HN-PWG [12]



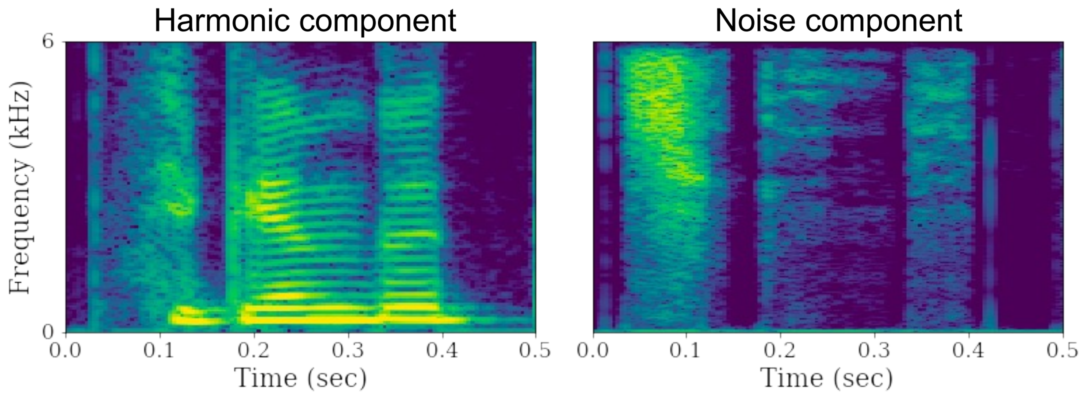
Step 4.
Sum all of subband signals

$$\mathbf{x} = \sum_{i=0}^{N-1} [\hat{\mathbf{x}}_{h,i} + \hat{\mathbf{x}}_{n,i}]$$

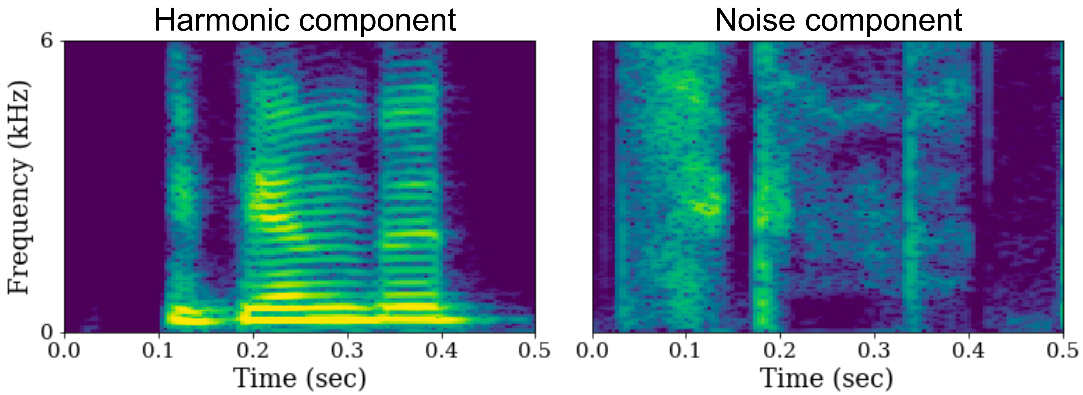
Multi-band HN-PWG vocoder

Spectrogram comparison with HN-PWG

- HN-PWG



- Multi-band HN-PWG



EXPERIMENTS

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	3.81	294.12	4.22	4.03
PWG	0.94	0.02	3.46	3.56
HN-PWG	0.94	0.02	4.18	4.01
Multi-band HN-PWG	0.99	0.02	4.29	4.03
Recordings	-	-	4.41	

EXPERIMENTS

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	**3.81**	**294.12**	4.22	4.03
PWG	**0.94**	**0.02**	3.46	3.56
HN-PWG	**0.94**	**0.02**	4.18	4.01
Multi-band HN-PWG	**0.99**	**0.02**	4.29	4.03
Recordings	-	-	4.41	

1. **Non-AR models provided significantly faster synthesis speed and smaller network size than AR-WaveNet.**

EXPERIMENTS

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	3.81	294.12	4.22	4.03
PWG	**0.94**	**0.02**	3.46	3.56
HN-PWG	**0.94**	**0.02**	4.18	4.01
Multi-band HN-PWG	**0.99**	**0.02**	4.29	4.03
Recordings	-	-	4.41	

1. Non-AR models provided significantly faster synthesis speed and smaller network size than AR-WaveNet.
2. **Use of HN model didn't affect the model size and inference speed.**

Experiments

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	3.81	294.12	**4.22**	**4.03**
PWG	0.94	0.02	**3.46**	**3.56**
HN-PWG	0.94	0.02	4.18	4.01
Multi-band HN-PWG	0.99	0.02	4.29	4.03
Recordings	-	-	4.41	

1. Non-AR models provided significantly faster synthesis speed and smaller network size than AR-WaveNet.
2. Use of HN model didn't affect the model size and inference speed.
3. **Conventional PWG showed worse quality than WaveNet.**

EXPERIMENTS

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	3.81	294.12	4.22	4.03
PWG	0.94	0.02	**3.46**	**3.56**
HN-PWG	0.94	0.02	**4.18**	**4.01**
Multi-band HN-PWG	0.99	0.02	**4.29**	**4.03**
Recordings	-	-	4.41	

1. Non-AR models provided significantly faster synthesis speed and smaller network size than AR-WaveNet.
2. Use of HN model didn't affect the model size and inference speed.
3. Conventional PWG showed worse quality than WaveNet.
4. **However, its quality was significantly improved by adopting HN model.**

EXPERIMENTS

Results

PWG: Parallel WaveGAN
HN-PWG: Harmonic-plus-noise PWG

Model	Model size ↓ (M)	Inference speed ↓ (RTF)	MOS ↑	
			Analysis / synthesis scenario	TTS scenario
WaveNet	3.81	294.12	**4.22**	**4.03**
PWG	0.94	0.02	3.46	3.56
HN-PWG	0.94	0.02	4.18	4.01
Multi-band HN-PWG	0.99	0.02	**4.29**	**4.03**
Recordings	-	-	4.41	

1. Non-AR models provided significantly faster synthesis speed and smaller network size than AR-WaveNet.
2. Use of HN model didn't affect the model size and inference speed.
3. Conventional PWG showed worse quality than WaveNet.
4. However, its quality was significantly improved by adopting HN model.
5. **Use of multi-band HN model improved quality of HN-PWG, and even better than AR WaveNet.**

Speech samples

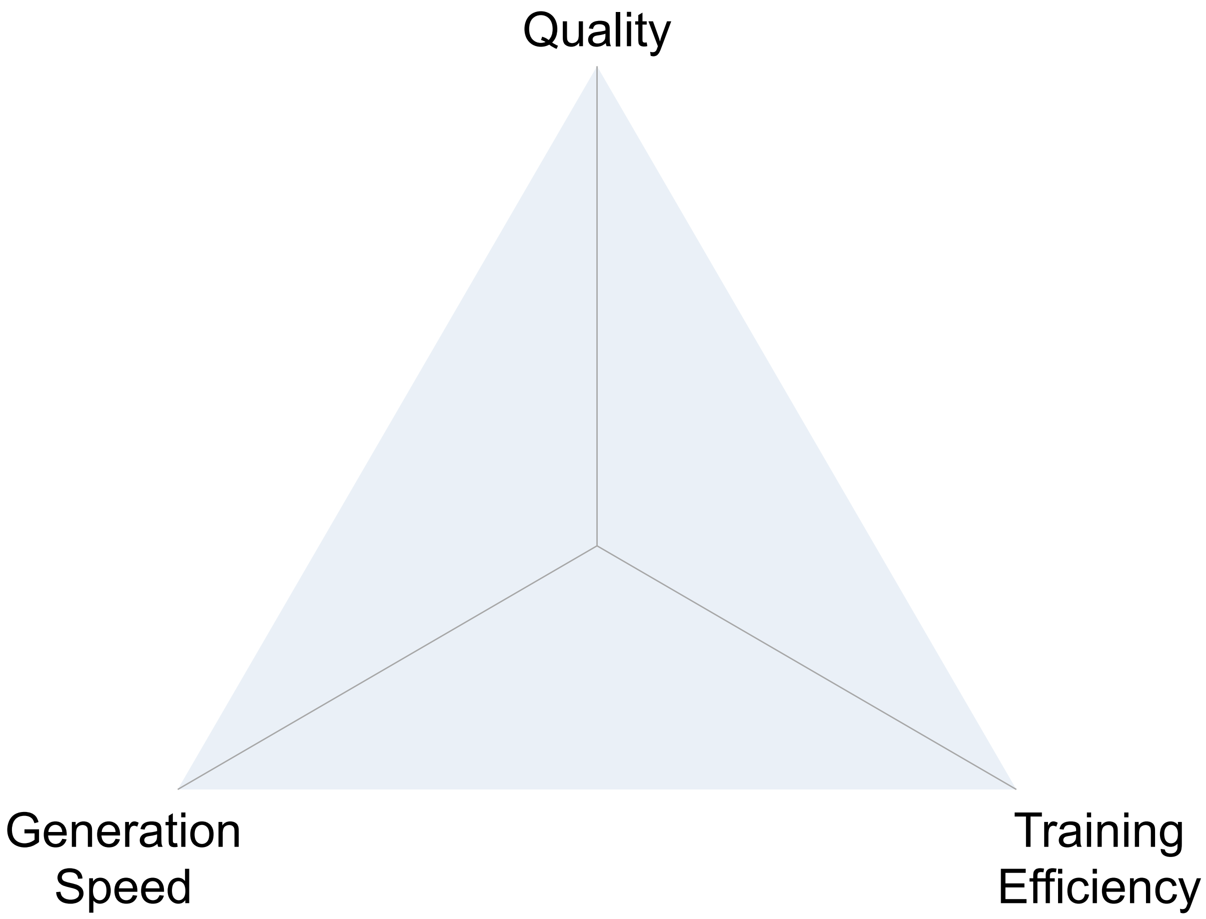
Recorded

HiFi-GAN [13]: state-of-the-art non-AR vocoder

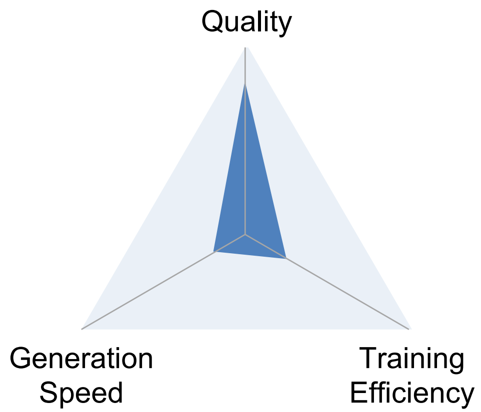
Multi-band HN-PWG (Analysis/synthesis)

Multi-band HN-PWG (TTS)

SUMMARY

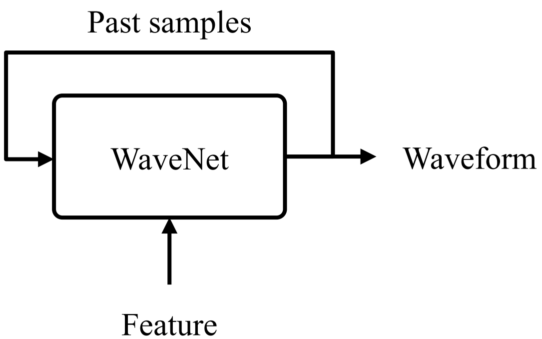


SUMMARY



WaveNet (MDN) [6]

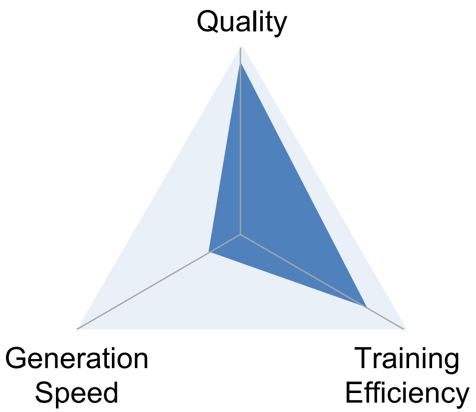
- First AR vocoder for speech waveform



[WaveNet]

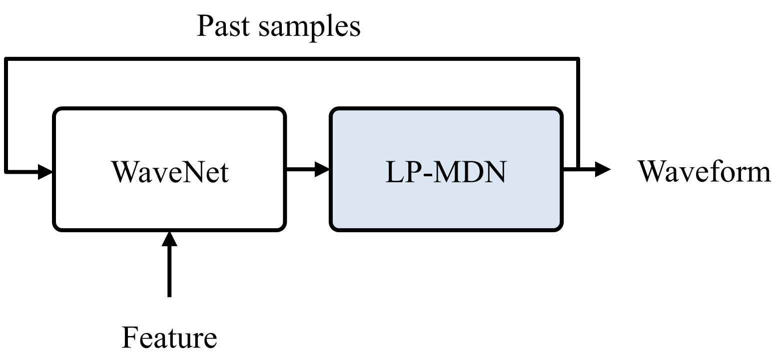
☺ Good quality
☹ Slow generation speed
☹ Difficult to train

SUMMARY



LP-WaveNet [9]

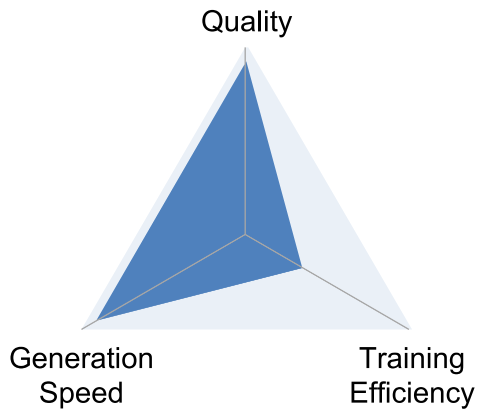
- Adopt LP-MDN to WaveNet



[LP-WaveNet]

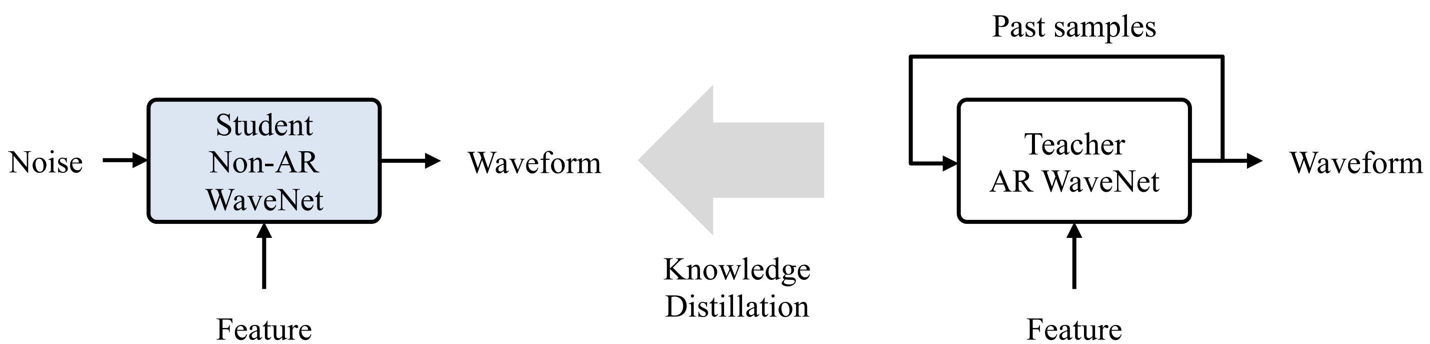
☺ Even better quality
☹ Slow generation speed
☺ Easy to train

SUMMARY



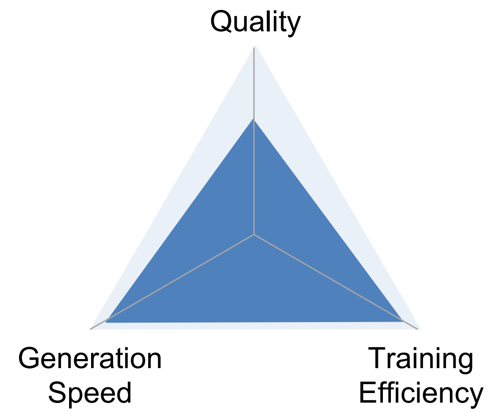
Parallel WaveNet [11]

- Non-AR WaveNet with teacher-student framework



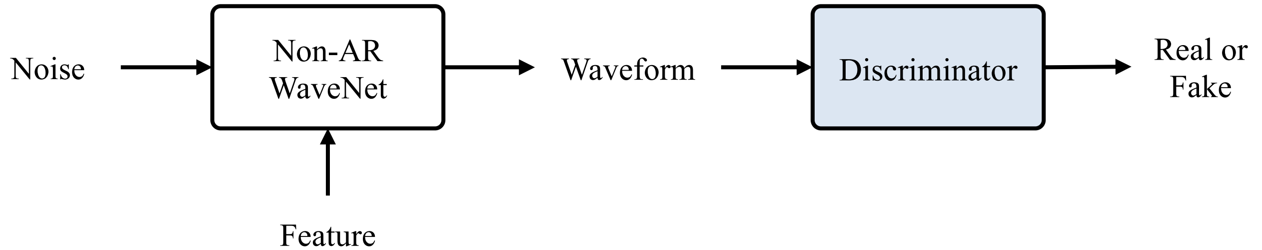
☺ Good quality
☺ Fast generation speed
☹ Too long training period

SUMMARY



Parallel WaveGAN (PWG) [4]

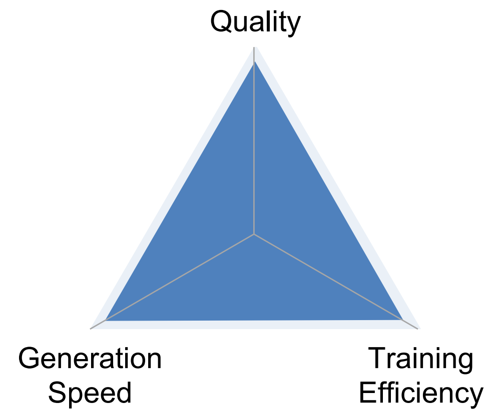
- Non-AR WaveNet with GAN framework



[PWG]

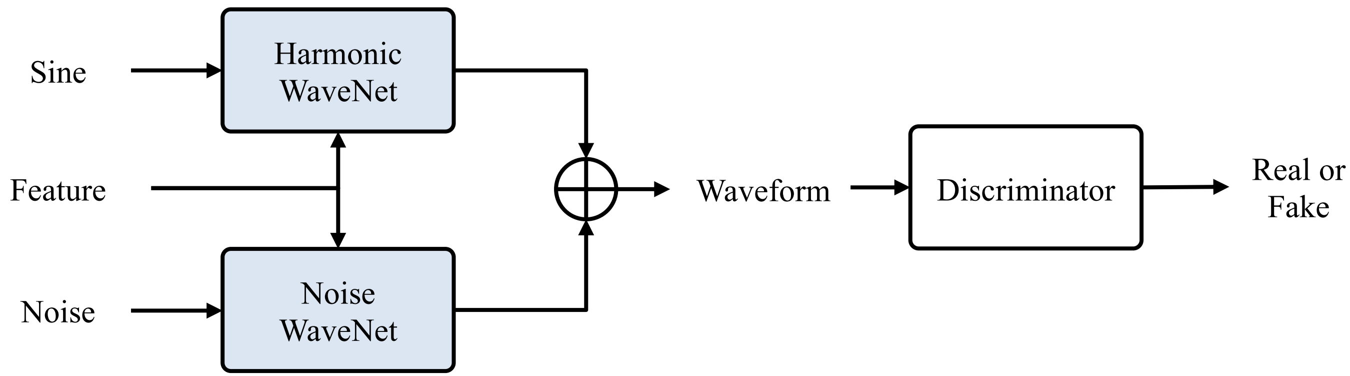
☹ Bad quality
☺ Fast generation speed
☺ Easy to train

SUMMARY



Harmonic-plus-noise PWG [13]

- Adopt HN model to PWG
- Proposed full-band and multi-band models



[HN-PWG]

☺ High quality
☺ Fast generation speed
☺ Easy to train

Replaced the role of LP-WaveNet, and applied to Naver's TTS services

SUMMARY

References

[1] Shen et al., “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in *Proc. ICASSP,* 2018.

[2] Ren et al., "FastSpeech: Fast, Robust and Controllable Text to Speech," in *Proc. NeurIPS*, 2019.

[3] Aaron et al., "WaveNet: A Generative Model for Raw Audio," in *Arxiv*, 2016.

[4] R. Yamamoto et al., "Parallel WaveGAN: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram," *in Proc. ICASSP,* 2020.

[5] A. Tamamori et al., “Speaker-dependent WaveNet vocoder,” in *Proc. Interspeech*, 2017.

[6] C. M. Bishop, “Mixture density networks,” Tech. Report, 1994.

[7] T. Quatieri, *Discrete-time Speech Signal Processing: Principles and Practice*. Prentice Hall Press, 2001.

[8] M.-J. Hwang et al., "LP-WaveNet: Linear Prediction-based WaveNet Speech Synthesis," in *Proc. APSIPA*, 2020.

[9] E. Song et al., “Effective spectral and excitation modeling techniques for LSTM-RNN-based speech synthesis systems,” in *IEEE/ACM Trans. ASLP*, 2017.

[10] A. van den Oord et al., “Parallel WaveNet: Fast high-fidelity speech synthesis,” in *Proc. ICML*, 2018.

[11] Y. Stylianou, “Modeling speech based on harmonic plus noise models,” in Nonlinear Speech Modeling and Applications. Springer Berlin Heidelberg, 2005.

[12] M.-J. Hwang et al., "High-fidelity Parallel WaveGAN with Multi-band Harmonic-plus-Noise Model," in *Proc. Interspeech*, 2021.

[13] J. Kong et al., “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis,” in Proc. NeurIPS, 2020.

Thank you!