Using vocoders to create training data for speech spoofing countermeasures

Xin Wang, Junichi Yamagishi
National Institute of Informatics
Contents

- Introduction
  - Text-to-speech synthesis
  - Speech anti-spoofing

- Method
  - Copy-synthesized data as spoofed data
  - Contrastive feature loss over bona fide and copy-synthesized data pair

- Experiment

- Summary
  - We use external training data. Please be careful when interpreting the results.
  - We take a data-driven approach. Apologize that we cannot precisely explain the model behavior.

Introduction
Text-to-speech synthesis

Input text

Text-to-speech synthesis

Marianna made the marmalade

Text-to-speech synthesis

Waveform generation

Acoustic realization

+Prosody tags

To phone

Normalization

Marianna made the marmalade


LOGIOS Lexicon tool: http://www.speech.cs.cmu.edu/tools/lextool.html

Text-to-speech synthesis – Unit selection

Waveform concatenation

Acoustic realization +Prosody tags To phone

Normalization

Marianna made the marmalade

Hunt, A. J. & Black, A. W. Unit selection in a concatenative speech synthesis system using a large speech database. in Proc. ICASSP 373–376 (1996).

Black, A. W. & Taylor, P. A. Automatically clustering similar units for unit selection in speech synthesis. (1997)
Text-to-speech synthesis – HMM

Marianna made the marmalade

Waveform generation → Vocoder

Acoustic realization

Hidden Markov model (HMM)

To phone

M A A R I Y AA N AH M E Y D D H A H M A A R M A H L E Y D

M A A R I Y AA N AH M E Y D D H A H M A A R M A H L E Y D

M A A R I Y AA N AH M E Y D D H A H M A A R M A H L E Y D

M A A R I Y AA N AH M E Y D D H A H M A A R M A H L E Y D

Normali- zation

M a ... made mar... lade

Normali- zation

M a ... made mar... lade

Marianna made the marmalade

HMM-Based Speech Synthesis Toolkit (HTS), home page: http://hts.sp.nitech.ac.jp/?Welcome
Text-to-speech synthesis

Waveform generation

Acoustic realization

+Prosody tags

To phonemes

Normalisation

Marianna made the marmalade


Text-to-speech synthesis – Recent methods


Xin Wang, Neural statistical parametric speech synthesis, ISCA Odyssey 2020, tutorial: https://tonywangx.github.io/slide.html#dec-2020
Voice conversion – Recent methods

- Neural vocoder
  - (Sequence-to-sequence)
  - Acoustic model
  - Feature extractor
  - Target speaker ID

Acoustic features

Source wav
Rapid progress of TTS

Waveform concatenation
Formant synthesis

Unit-selection
~1996

HTS
~2000

HMM-DNN
~2013

WaveNet Seq-to-seq
~2016

Now

<table>
<thead>
<tr>
<th>Unit-selection</th>
<th>HMM-DNN (close to HMM)</th>
<th>HMM-DNN (close to DNN)</th>
<th>Seq2seq</th>
<th>Natural speech</th>
</tr>
</thead>
</table>

More samples: Xin Wang, Neural statistical parametric speech synthesis, ISCA Odyssey 2020, tutorial: https://tonywax.github.io/slide.html#dec-2020
TTS/VC may be misused

Grandma, this is **PAUL**. Can you please send $500 …
TTS/VC may be misused

Fraudsters Used AI to Mimic CEO’s Voice in Unusual Cybercrime Case

Scams using artificial intelligence are a new challenge for companies.
Spoofing countermeasure

Human users → TTS

Spoofing countermeasure (CM)

Attackers

Valid

Users
Smart phones
Organizations
Online services
Call center
Spoofing countermeasure

- Model construction
  - a binary classification model

- Model evaluation
  - accuracy
  - equal error rate (EER)
  - tandem detection cost function (t-DCF) \( (\text{Kinnunen 2020}) \)

[Diagram showing feature extraction, scoring, and decision process]

- Input wav. → Feature extraction (front end) → Scoring (back end)
  - CM score \( s \) > \( \theta_s \) → Bona fide
  - CM score \( s \) < \( \theta_s \) → Spoofed

---

Speech anti-spoofing

- training and test sets from a database
  - ASVspoof [www.asvspoof.org](http://www.asvspoof.org)
  - BTAS2016
  - FMFCC-A
  - ...

Space of all possible bona fide and spoofed data

<table>
<thead>
<tr>
<th>Training set</th>
<th>CM</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Training set image" /></td>
<td><img src="image2.png" alt="CM image" /></td>
<td><img src="image3.png" alt="Test set image" /></td>
</tr>
</tbody>
</table>

En, Fr, Ch, Jp, …

Wav, mp3, m4a …

New TTS/VC methods
Speech anti-spoofing

Just 25k utterances from 20 speakers!?

- poor generalization (Das 2020)
- dataset-specific bias?
  - biased dist. of non-speech (Müller 2021, Liu 2022)
  - artefacts in high-frequency band (Wang 2022)


“Dead” data in self-supervised speech model (Xie 2021, Wang 2022, Tak 2022, Donas 2022)

Can we add more diverse training data?

Space of all possible bona fide and spoofed data

Training set

Test set

New TTS/VC methods

En, Fr, Ch, Jp, …

Wav, mp3, m4a …
Speech anti-spoofing

- Building various TTS and VC systems are time-consuming.
- We need some efficient ways to create spoofed training data

- “Dead” data in self-supervised speech model
  (Xie 2021, Wang 2022, Tak 2022, Donas 2022)

- Can we add more diverse training data?

Space of all possible bona fide and spoofed data

New TTS/VC methods

Training set

Test set

Wav, mp3, m4a …


Method
Copy synthesis

Acoustic features

Vocoder

Input text

a  ...  made  mar...lade
Copy synthesis

Acoustic features

Vocoder

In HMM & DNN era
- Mel cepstrum
- F0
- band aperiodicity

In latest DNN era
- spectrum, Mel spectrum, …
Copy synthesis

Vocoder

Acoustic feature extractor (usually automatic and deterministic)

Copy-synthesis, analysis-by-synthesis, copy-resynthesis, vocoding …
Copy synthesis (in history)

Copy-synthesis, analysis-by-synthesis, resynthesis, ...

Formant synthesizer (deterministic)

Acoustic feature extractor (manual w/ trial & error)

IV. Synthesis of Copies of Natural Utterances

preset bandwidth values. The techniques used are mostly of the “analysis-by-synthesis” type, with a human interpreter of differences between natural and synthetic speech in the feedback loop.


We do copy-synthesis when training the neural vocoders
Creating copy-synthesized spoofed data

Steps

- prepare (or training) vocoders
  - not necessarily neural vocoders
- Do copy synthesis on bona fide data
- use output as copy-synthesized (or vocoded) spoofed data
- train a CM using \{bona fide, copy-synthesized spoofed\}
Creating copy-synthesized spoofed data

- **Hypothesis**
  - *Copy-synthesis is TTS/VC with a perfect acoustic model*
    - × artefacts by the acoustic model
    - ✓ artefacts by the vocoder
  - TTS/VC spoofed data may contain artefacts by the vocoder

WaveGlow “bar” (Prenger 2019)  
WaveNet “click” (Wu 2018)  

---


Creating copy-synthesized spoofed data

Questions

- How to prepare or train the vocoder?
  - pre-trained vocoder(s)?
  - Fine tuning?

- Can we better use the aligned bona fide and spoofed data pairs?

Experiment I

Experiment II
Creating copy-synthesized spoofed data

Related studies using DSP-based vocoders


Related studies using neural vocoders

Experiment I
How to prepare or train the vocoder?
Experiment I

- **Design**

- **Training**
  - changed factor: training data created by different sets of vocoders
  - unchanged: CM (Wang 2022) using a Wav2vec2.0-based front end (Baevski 2020)
  - unchanged: multiple test sets

- **Evaluation**

- three independent training & evaluation rounds
- averaged EER
Experiment I

- Design

- changed factor: training data created by different sets of vocoders
- unchanged: CM (Wang 2022) using a Wav2vec2.0-based front end (Baevski 2020)
- unchanged: multiple test sets
  - ASVspoof 2019 LA test set w/o non-speech, 2021 LA & DF hidden track
  - WaveFake (Frank 2021), In-the-Wild (Müller 2022)
Design

Original test trials

Non-speech trimmed test trials

(figure from Zhang 2021)

• ASVspoof 2019 LA test set w/o non-speech, 2021 LA & DF hidden track

We recommend testing on both versions of ASVspoof test sets
## Experiment I

### Training data

<table>
<thead>
<tr>
<th>ID</th>
<th>#. Spr.</th>
<th>#. Bona.</th>
<th>#. Spoof.</th>
<th>Vocoder type</th>
<th>Implementation</th>
<th>Vocoder train/fine-tune data</th>
<th>Vocoder SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19trn</td>
<td>20</td>
<td>2,580</td>
<td>22,800</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16 kHz</td>
</tr>
<tr>
<td>WFtrn</td>
<td>1</td>
<td>3,930</td>
<td>15,720</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v1</td>
<td>20</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, MB-MelGAN, PWG, StyleMelGAN</td>
<td>ESPNet toolkit</td>
<td>LibriTTS / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v2</td>
<td>same as LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v3</td>
<td>LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v4</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / LA19trn bona.</td>
<td>16 kHz</td>
</tr>
</tbody>
</table>

- **LA19trn**: ASVspoof 2019 LA training set
- **WFtrn**: WaveFake English subset, down-sampled to 16kHz
- **Voc.v***: ASVspoof 2019 LA training set bona fide data + **its vocoded data**
## Experiment I

- **Results in EER (%)**

<table>
<thead>
<tr>
<th>Test sets</th>
<th>LA19</th>
<th>WF</th>
<th>Voc. v1</th>
<th>Voc. v2</th>
<th>Voc. v3</th>
<th>Voc. v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td>44.48</td>
<td>5.78</td>
<td>5.32</td>
<td>8.74</td>
<td>4.36</td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td>41.57</td>
<td>26.30</td>
<td>17.98</td>
<td>19.29</td>
<td>24.39</td>
</tr>
<tr>
<td>DF21eval</td>
<td>6.67</td>
<td>24.26</td>
<td>11.95</td>
<td>11.54</td>
<td>9.71</td>
<td>13.31</td>
</tr>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>31.62</td>
<td>23.29</td>
<td>16.16</td>
<td>14.99</td>
<td>9.52</td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>27.60</td>
<td>28.30</td>
<td>19.49</td>
<td>17.62</td>
<td>21.43</td>
</tr>
<tr>
<td>WaveFake</td>
<td>15.76</td>
<td>-</td>
<td>39.27</td>
<td>34.05</td>
<td>17.10</td>
<td>10.89</td>
</tr>
<tr>
<td>InWild</td>
<td>26.65</td>
<td>19.98</td>
<td>41.06</td>
<td>36.46</td>
<td>22.26</td>
<td>19.45</td>
</tr>
<tr>
<td><strong>Pooled</strong></td>
<td>14.24</td>
<td>-</td>
<td>36.57</td>
<td>39.95</td>
<td>19.39</td>
<td>16.35</td>
</tr>
</tbody>
</table>

### Single threshold

- **Low EER**
- **High EER**
## Experiment I

### Results in EER (%, mean of three runs)

<table>
<thead>
<tr>
<th>Test sets</th>
<th>LA19etr</th>
<th>WF et</th>
<th>Voc v1</th>
<th>Voc v2</th>
<th>Voc v3</th>
<th>Voc v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td>44.48</td>
<td>5.78</td>
<td>5.32</td>
<td>8.74</td>
<td>4.36</td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td>41.82</td>
<td>11.95</td>
<td>11.54</td>
<td>9.71</td>
<td>13.31</td>
</tr>
<tr>
<td>DF21eval</td>
<td>6.67</td>
<td>24.29</td>
<td>22.01</td>
<td>13.92</td>
<td>13.50</td>
<td>16.99</td>
</tr>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>31.02</td>
<td>23.29</td>
<td>16.10</td>
<td>14.99</td>
<td>9.02</td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>27.60</td>
<td>28.30</td>
<td>19.49</td>
<td>17.62</td>
<td>21.43</td>
</tr>
<tr>
<td>WaveFake</td>
<td>15.76</td>
<td>19.35</td>
<td>34.05</td>
<td>17.10</td>
<td>10.89</td>
<td></td>
</tr>
<tr>
<td>InWild</td>
<td>26.65</td>
<td>19.38</td>
<td>36.46</td>
<td>22.26</td>
<td>19.45</td>
<td></td>
</tr>
</tbody>
</table>

*Similar results to our previous work (Wang 2022)*

*Same trend as previous studies (Müller 2021, Liu 2022)*

*Not 50% :)*
### Results in EER (%, mean of three runs)

#### Experiment I

<table>
<thead>
<tr>
<th>ID</th>
<th>#. Spr.</th>
<th>#. Bona.</th>
<th>#. Spoof.</th>
<th>Vocoder type</th>
<th>Implementation</th>
<th>Vocoder train/fine-tune data</th>
<th>Vocoder SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19trn</td>
<td>20</td>
<td>2,580</td>
<td>22,800</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16 kHz</td>
</tr>
<tr>
<td>WFtrn</td>
<td>1</td>
<td>3,930</td>
<td>15,720</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v1</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, MB-MelGAN, PWG, StyleMelGAN</td>
<td>ESPNet toolkit</td>
<td>LibriTTS / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v2</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v3</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v4</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / LA19trn bona.</td>
<td>16 kHz</td>
</tr>
</tbody>
</table>

#### Training set

<table>
<thead>
<tr>
<th></th>
<th>LA19trn</th>
<th>WFtrn</th>
<th>Voc. v1</th>
<th>Voc. v2</th>
<th>Voc. v3</th>
<th>Voc. v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td>44.48</td>
<td>5.78</td>
<td>5.32</td>
<td>8.74</td>
<td>4.36</td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td>41.57</td>
<td>26.30</td>
<td>17.98</td>
<td>19.29</td>
<td>24.39</td>
</tr>
<tr>
<td>DF21eval</td>
<td>6.67</td>
<td>24.26</td>
<td>11.95</td>
<td>11.54</td>
<td>9.71</td>
<td>13.31</td>
</tr>
</tbody>
</table>

#### Test sets

<table>
<thead>
<tr>
<th></th>
<th>LA19etrim</th>
<th>WF</th>
<th>Voc. v1</th>
<th>Voc. v2</th>
<th>Voc. v3</th>
<th>Voc. v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>31.62</td>
<td>23.29</td>
<td>18.02</td>
<td>13.50</td>
<td>16.99</td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>27.60</td>
<td>28.30</td>
<td>13.05</td>
<td>17.10</td>
<td>10.89</td>
</tr>
<tr>
<td>DF21hid</td>
<td>23.62</td>
<td>26.18</td>
<td>22.01</td>
<td>36.46</td>
<td>22.26</td>
<td>19.45</td>
</tr>
<tr>
<td>WaveFake</td>
<td>15.76</td>
<td>-</td>
<td>39.27</td>
<td>22.26</td>
<td>19.45</td>
<td>16.35</td>
</tr>
<tr>
<td>InWild</td>
<td>26.65</td>
<td>19.98</td>
<td>41.06</td>
<td>39.95</td>
<td>19.39</td>
<td>16.35</td>
</tr>
</tbody>
</table>

#### Vocoder type

- WaveFake
- InWild
- LA19etrim
- LA21hid
- DF21hid

#### Vocoder pre-trained by ESPNet (Hayashi 2020)
<table>
<thead>
<tr>
<th>ID</th>
<th>#. Spr.</th>
<th>#. Bona.</th>
<th>#. Spoof.</th>
<th>Vocoder type</th>
<th>Implementation</th>
<th>Vocoder train/fine-tune data</th>
<th>Vocoder SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19trn</td>
<td>20</td>
<td>2,580</td>
<td>22,800</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WFtrn</td>
<td>1</td>
<td>3,930</td>
<td>15,720</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v1</td>
<td>20</td>
<td>same as LA19trn</td>
<td></td>
<td>HiFiGAN, MB-MelGAN, PWG, StyleMelGAN</td>
<td>ESPNet toolkit</td>
<td>LibriTTS / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v2</td>
<td>20</td>
<td>same as LA19trn</td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v3</td>
<td>20</td>
<td>same as LA19trn</td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v4</td>
<td>20</td>
<td>same as LA19trn</td>
<td></td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / LA19trn bona.</td>
<td>16 kHz</td>
</tr>
</tbody>
</table>

### Results in EER (%, mean of three runs)

#### Experiment I

<table>
<thead>
<tr>
<th>Training set</th>
<th>LA19trn</th>
<th>WFtrn</th>
<th>Voc. v1</th>
<th>Voc. v2</th>
<th>Voc. v3</th>
<th>Voc. v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td>44.48</td>
<td>5.78</td>
<td>5.32</td>
<td>8.74</td>
<td>4.36</td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td>41.57</td>
<td>26.30</td>
<td>17.98</td>
<td>19.29</td>
<td>24.39</td>
</tr>
<tr>
<td>DF21eval</td>
<td>6.67</td>
<td>24.26</td>
<td>11.95</td>
<td>11.54</td>
<td>9.71</td>
<td>13.31</td>
</tr>
</tbody>
</table>

#### Test sets

<table>
<thead>
<tr>
<th>Test sets</th>
<th>LA19etrim</th>
<th>LA21hid</th>
<th>DF21hid</th>
<th>WaveFake</th>
<th>InWild</th>
<th>Pooled</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>31.62</td>
<td>23.29</td>
<td>16.16</td>
<td>14.99</td>
<td>9.52</td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>27.60</td>
<td>28.30</td>
<td>19.49</td>
<td>17.62</td>
<td>21.43</td>
</tr>
<tr>
<td>WaveFake</td>
<td>15.76</td>
<td>-</td>
<td>39.27</td>
<td>34.05</td>
<td>17.10</td>
<td>10.89</td>
</tr>
<tr>
<td>InWild</td>
<td>26.65</td>
<td>19.98</td>
<td>41.06</td>
<td>36.46</td>
<td>22.26</td>
<td>19.45</td>
</tr>
</tbody>
</table>

**Was vocoder trained on the bona fide data?** No
<table>
<thead>
<tr>
<th>ID</th>
<th>#. Spr.</th>
<th>#. Bona.</th>
<th>#. Spoof.</th>
<th>Vocoder type</th>
<th>Implementation</th>
<th>Vocoder train/fine-tune data</th>
<th>Vocoder SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19trn</td>
<td>20</td>
<td>2,580</td>
<td>22,800</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>WFtrn</td>
<td>1</td>
<td>3,930</td>
<td>15,720</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v1</td>
<td>20</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, MB-MelGAN, PWG, StyleMelGAN</td>
<td>ESPNet toolkit</td>
<td>LibriTTS / -</td>
<td>24 kHz</td>
</tr>
<tr>
<td>Voc.v2</td>
<td>same as LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v3</td>
<td>LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>16 kHz</td>
</tr>
<tr>
<td>Voc.v4</td>
<td>LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>16 kHz</td>
</tr>
</tbody>
</table>

**Training set**

<table>
<thead>
<tr>
<th></th>
<th>LA19trn</th>
<th>WFtrn</th>
<th>Voc. v1</th>
<th>Voc. v2</th>
<th>Voc. v3</th>
<th>Voc. v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td></td>
<td>2.98</td>
<td>3.92</td>
<td>8.74</td>
<td>4.36</td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td></td>
<td>7.53</td>
<td>5.32</td>
<td>8.74</td>
<td>24.39</td>
</tr>
<tr>
<td>DF21eval</td>
<td>6.67</td>
<td></td>
<td>6.67</td>
<td>5.32</td>
<td>8.74</td>
<td>13.31</td>
</tr>
</tbody>
</table>

**Test sets**

<table>
<thead>
<tr>
<th></th>
<th>LA19etrim</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>15.56</td>
<td>9.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>28.80</td>
<td>21.43</td>
<td>16.99</td>
<td></td>
</tr>
<tr>
<td>DF21hid</td>
<td>23.62</td>
<td>23.62</td>
<td>16.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WaveFake</td>
<td>15.76</td>
<td>15.76</td>
<td>10.89</td>
<td>19.45</td>
<td></td>
</tr>
<tr>
<td>InWild</td>
<td>26.65</td>
<td>26.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pooled</td>
<td>14.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We cannot exploit

non-speech length

Reasonably good
Experiment I

- How to prepare or train the vocoder?
  - Pre-trained vocoders may not work --- WFtrn, Voc.v1
  - It is better to fine tune vocoders on the bona fide to be copy-synthesized

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LA19trn</td>
<td>25</td>
<td>2,580</td>
<td>22,800</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>16k</td>
</tr>
<tr>
<td>WFtrn</td>
<td>1</td>
<td>3,930</td>
<td>15,720</td>
<td>HiFiGAN, MB-MelGAN, PWG, WaveGlow</td>
<td>ESPNet toolkit</td>
<td>LJSpeech / -</td>
<td>24k</td>
</tr>
<tr>
<td>Voc.v1</td>
<td></td>
<td></td>
<td></td>
<td>HiFiGAN, MB-MelGAN, PWG, StyleMelGAN</td>
<td>ESPNet toolkit</td>
<td>LibriTTS / -</td>
<td>24k</td>
</tr>
<tr>
<td>Voc.v2</td>
<td>20</td>
<td></td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / -</td>
<td>16k</td>
</tr>
<tr>
<td>Voc.v3</td>
<td>same</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LA19trn bona. / -</td>
<td>16k</td>
</tr>
<tr>
<td>Voc.v4</td>
<td>LA19trn</td>
<td>2,580</td>
<td>10,320</td>
<td>HiFiGAN, NSF-HiFiGAN, Hn-NSF, WaveGlow</td>
<td>in-house</td>
<td>LibriTTS / LA19trn bona.</td>
<td>16k</td>
</tr>
</tbody>
</table>

- CAUTION: other CMs (e.g., LCNN and AASIST) did not work well on vocoded data
Experiment II
How to make good use of the aligned bona fide and vocoded data pair?

\[ \hat{o}_{1:T} \]

\[ o_{1:T} \]
Experiment II

- **Method:** make use of the aligned bona fide and vocoded pair
  - Existing method: use their differences in frequency domain \(^{(Wang \ 2021)}\)
    - Vocoders are needed during inference
    - Too slow to score the test sets
  - We proposed an auxiliary contrastive feature loss \(^{(Khosla \ 2020)}\)
    - It is based on supervised contrastive loss
    - It needs data augmentation (DA)
    - No need to run vocoders during inference
Experiment II

- **Method:** an auxiliary contrastive feature loss

![Diagram](https://example.com/diagram.png)

Cross entropy loss: $\mathcal{L}_{CE}$

Contrastive feature loss: $\mathcal{L}_{CF}$

Cosine similarity over sequences:

- $\log \frac{\text{Sim}(\ldots)}{\text{Sim}(\ldots) + \text{Sim}(\ldots) + \text{Sim}(\ldots)}$
- $\log \frac{\text{Sim}(\ldots)}{\text{Sim}(\ldots) + \text{Sim}(\ldots) + \text{Sim}(\ldots)}$

Experiment II

- Design

  - unchanged: training data **Voc.v4**
  - unchanged: same CM architecture as Experiment I
  - unchanged: multiple test sets

  - changed: training criterion and method
    - data augmentation based on RawBoost (Tak 2022)
## Experiment II

### Results

<table>
<thead>
<tr>
<th>Training criterion</th>
<th>$\mathcal{L}_{\text{CE}}$</th>
<th>$\mathcal{L}<em>{\text{CE}} + \mathcal{L}</em>{\text{CF}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data augmentation</td>
<td>$\times$</td>
<td>RawBoost</td>
</tr>
<tr>
<td>Training set</td>
<td></td>
<td>RawBoost</td>
</tr>
<tr>
<td>Bona-spoof paired</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>ID</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Training sets**

- **LA19eval**: 2.98 (4.36) 0.22 (3.46) 0.21 (2.63) 2.21
- **LA21eval**: 7.53 (24.39) 3.63 (16.55) 3.30 (16.67) 17.90
- **DF21eval**: 6.67 (13.31) 3.65 (9.60) 4.42 (6.92) 5.04

**Test sets**

- **LA19etrim**: 15.56 (9.52) 9.16 (6.09) 9.00 (4.48) 3.79
- **LA21hid**: 28.80 (21.43) 21.18 (19.37) 26.98 (15.05) 14.57
- **DF21hid**: 23.62 (16.99) 13.64 (14.29) 16.85 (8.17) 7.78
- **WaveFake**: 15.76 (10.89) 26.37 (6.87) 24.62 (4.03) 2.50
- **InWild**: 26.65 (19.45) 16.17 (12.08) 17.07 (9.37) 7.55
- **Pooled**: 14.24 (16.35) 13.12 (13.13) 13.68 (13.15) 11.27

**Control groups**

- **Best**

**Results from Experiment I**

- 2 vs 4: RawBoost is useful
- 4 vs 7: Contrastive feature loss is useful
## Experiment II

### Results

from Experiment I  
Control groups  
Best

<table>
<thead>
<tr>
<th>Training criterion</th>
<th>$\mathcal{L}_{CE}$</th>
<th>$\mathcal{L}<em>{CE} + \mathcal{L}</em>{CF}$</th>
<th>RawBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data augmentation</td>
<td>$\times$</td>
<td>RawBoost</td>
<td></td>
</tr>
<tr>
<td>Training set</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>Bona-spoof paired</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LA19eval</td>
<td>2.98</td>
<td>4.36</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>3.66</td>
<td>3.46</td>
<td></td>
</tr>
<tr>
<td>LA21eval</td>
<td>7.53</td>
<td>24.39</td>
<td>3.63</td>
</tr>
<tr>
<td></td>
<td>6.67</td>
<td>16.55</td>
<td>4.12</td>
</tr>
<tr>
<td>DF21eval</td>
<td>3.67</td>
<td>13.31</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>13.12</td>
<td>9.60</td>
<td>6.92</td>
</tr>
<tr>
<td>LA19etrim</td>
<td>15.56</td>
<td>9.52</td>
<td>9.16</td>
</tr>
<tr>
<td></td>
<td>15.06</td>
<td>6.09</td>
<td></td>
</tr>
<tr>
<td>LA21hid</td>
<td>28.80</td>
<td>21.43</td>
<td>21.18</td>
</tr>
<tr>
<td></td>
<td>23.62</td>
<td>16.99</td>
<td>13.64</td>
</tr>
<tr>
<td>DF21hid</td>
<td>15.76</td>
<td>10.89</td>
<td>26.37</td>
</tr>
<tr>
<td></td>
<td>26.65</td>
<td>19.45</td>
<td>16.17</td>
</tr>
<tr>
<td></td>
<td>13.68</td>
<td>12.08</td>
<td></td>
</tr>
<tr>
<td>InWild</td>
<td>14.24</td>
<td>16.35</td>
<td>13.12</td>
</tr>
<tr>
<td></td>
<td>13.68</td>
<td>11.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13.68</td>
<td>11.27</td>
<td></td>
</tr>
</tbody>
</table>

② vs ④  
RawBoost is useful

④ vs ⑦  
Contrastive feature loss is useful

⑥ vs ⑦  
Use aligned {bona, vocoded} pairs!
Summary
Summary

- **Spoofed training data can be created using neural vocoders**

- **Recommendations (from this study)**
  - Fine-tune vocoders on the bona fide data to be copy-synthesized

  - Exploit the the aligned {bona fide, vocoded} pairs
    - for example, by using contrastive feature loss

  - The best trained CM showed promising generalization performance
Summary

- **Additional findings (see [https://arxiv.org/abs/2210.10570](https://arxiv.org/abs/2210.10570])**
  - Do we generalize to old TTS and VC?
    - Yes, EER on the ASVspoof 2015 test set < 1%
  - Is the CM sufficiently generalizable?
    - No, TTS/VC with autoregressive (AR) vocoders are challenging

![Score distributions](image)

**Fig. 1**: Score distributions of 📈 on *bona fide* and *spoofed* trials in DF21hid. Number in each sub-figure is EER (%).
In progress

- Scale to large data
  - CM: same as Experiment II
  - ASVspoof 2019 LA trn bona fide + vocoded data: 2 * 5 hours
  - VoxCeleb2 dev set + vocoded data: 2300 * 5 hours

<table>
<thead>
<tr>
<th>Training steps</th>
<th>ASVspoof 2019 trn</th>
<th>VoxCeleb2 dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N 0.3N 0.6N 1.2N 2.4N 6.0N 12.0N</td>
<td></td>
</tr>
<tr>
<td>LA19eval</td>
<td>2.21 8.20 7.04 5.40 6.53 5.40 5.63</td>
<td></td>
</tr>
<tr>
<td>LA21eval</td>
<td>17.90 20.19 16.73 14.33 18.10 17.44 17.84</td>
<td></td>
</tr>
<tr>
<td>DF21eval</td>
<td>5.04 7.49 5.41 5.39 6.00 5.65 5.64</td>
<td></td>
</tr>
<tr>
<td>LA19etrim</td>
<td>3.79 6.17 5.53 5.16 5.25 5.22 5.14</td>
<td></td>
</tr>
<tr>
<td>LA21hid</td>
<td>14.57 13.98 12.34 11.47 11.64 11.37 11.43</td>
<td></td>
</tr>
<tr>
<td>DF21hid</td>
<td>7.78 11.02 9.71 9.90 10.05 9.99 10.04</td>
<td></td>
</tr>
<tr>
<td>WaveFake</td>
<td>2.50 14.94 10.39 8.38 5.52 4.88 4.94</td>
<td></td>
</tr>
<tr>
<td>InWild</td>
<td>7.55 16.12 15.63 14.19 13.43 13.32 13.77</td>
<td></td>
</tr>
<tr>
<td>Pooled</td>
<td>11.27 13.52 11.79 9.98 9.01 8.36 8.37</td>
<td></td>
</tr>
</tbody>
</table>
Resources

- Code, vocoded data, vocoders, and trained CMs [Git]

- Tutorials on neural vocoders (AR, flow, GAN, DSP …) [Git]
  - Jupyter notebooks & pre-trained models

- ASVspoof 2021 hidden track
  - They are already in the data packages you’ve downloaded
  - You just need to find them using the official meta labels
  - See https://github.com/asvspoof-challenge/2021
Thank you!
Appendix
Why TTS is difficult?

Speaker A: Who made the marmalade.

Speaker A: Bob made the marmalade.

Speaker B: (No,) Marianna made the marmalade.

Speaker B: Marianna made the marmalade.
CM architecture

Input wav. → Jointed tuned

wav2vec2.0
+ linear (128 dims)

Global Average pooling
+ MLP (dropout)
+ softmax

$S > \theta_s$ → Bona fide

$S < \theta_s$ → Spoofed

Pre-trained wav2vec 2.0 model
(XLSR-53, pre-trained on multi-lingual data)
Analysis

System 7 on LA 2021

System 7 on DF 2021