VoiceFilter-Lite: Streaming Targeted Voice Separation for On-Device Speech Recognition

Authors: Quan Wang, Ignacio Lopez Moreno, Mert Saglam, Kevin Wilson, Alan Chiao, Renjie Liu, Yanzhang He, Wei Li, Jason Pelecanos, Marily Nika, Alexander Gruenstein

Presented by: Quan Wang <quanw@google.com>
Key messages

● What:
  ○ A single-channel source separation model for a target speaker
  ○ Part of on-device streaming ASR

● Why:
  ○ Improve ASR on overlapped speech drastically

● How:
  ○ Filterbanks as inputs and outputs
  ○ Asymmetric loss and adaptive runtime suppression strength
  ○ Quantize to 8-bit integer model
Part 1:

Recap of VoiceFilter
The Cocktail Party Problem

The problem:

- Multiple talkers speaking simultaneously
- You just want to listen to one person

Conventional solutions:

- Multi-channel blind separation
- Single-channel blind separation
Knowing “whom to listen to”

- Voice Match already deployed to:
  - Android smart phones
  - Smart home speakers
- User’s **voice profile** stored on device

- Blindness sucks
  - **Number of sources** is unknown
    - Single: don’t want to mess up
    - Multiple: only keep target voice
  - **Multiple outputs**, still don’t know which one to use (e.g. to run ASR)
    - Pick loudest?
    - Run speaker verification on all?
  - **Permutation invariance**: computational cost is high!
VoiceFilter: Say Goodbye to blindness!

- Condition the voice separation task by target speaker embedding

Noisy multi-speaker audio → VoiceFilter → Clean audio of target speaker
Learn more about VoiceFilter

- https://www.youtube.com/watch?v=gnRX2lzepz0
Part 2:

VoiceFilter for on-device ASR
On-device ASR

● Moving ASR from cloud to device is the trend
  ○ No requirement for **Internet**
  ○ Much less **latency** due to communications
  ○ Better **privacy** preservation

● Example use cases (smartphones, smart home devices):
  ○ “Turn on flashlight”
  ○ “Turn on bedroom lights”
Challenges: On-device VoiceFilter

- **Memory and storage:**
  - Model should be really tiny (few MB, not GB)

- **CPU and battery:**
  - Less runtime operations

- **Latency:**
  - Model should work in a streaming fashion
  - ASR should never wait for VoiceFilter
Challenges: VoiceFilter for ASR

- Quality:
  - Being always harmless is more important than being sometimes helpful!
  - Clean single speaker
  - Non-speech noise
  - Reverberant rooms
  - Different SNR

- VoiceFilter: You either always run it, or never run it
Part 3:

The journey to Lite
Optimizing for CPU and latency

- Time-frequency CNN
  - Temporal CNN: Not causal
  - Convolutions: Computationally expensive
Optimizing for CPU and latency

- Bi-directional LSTM
  - Backward pass: Not causal

We can't use backward LSTM
Optimizing for CPU and latency

- If we only care about ASR:
  - ASR inputs directly as VoiceFilter inputs & outputs
  - No need to convert back to waveform
  - No cool audio demos
Let’s simplify the feature frontend

- ASR uses stacked log Mel-filterbanks as input
Model architecture

- Assume we use stacked filterbanks as VoiceFilter-Lite I/O
## What have changed since VoiceFilter

<table>
<thead>
<tr>
<th></th>
<th>VoiceFilter</th>
<th>VoiceFilter-Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input &amp; Output</strong></td>
<td>Audio waveform</td>
<td>ASR features (FFT magnitude, filterbank, etc.)</td>
</tr>
<tr>
<td><strong>Inference</strong></td>
<td>Full sequence offline inference</td>
<td>Online streaming inference (no time-freq CNN, no bi-LSTM)</td>
</tr>
<tr>
<td><strong>STFT and iSTFT</strong></td>
<td>Part of VoiceFilter model graph</td>
<td>STFT as part of ASR system; no iSTFT</td>
</tr>
<tr>
<td><strong>Model format and size</strong></td>
<td>TensorFlow graph; typically &gt;100MB</td>
<td>Quantized to int8 TFLite model; typically &lt;10MB</td>
</tr>
</tbody>
</table>
VoiceFilter-Lite vs joint training with ASR

- **Joint training with ASR**
  - Concatenate d-vector with filterbanks for ASR training
  - Creates dependency between ASR and speaker recognition models
  - What if user did not enroll (d-vector not available)?

- **VoiceFilter-Lite**
  - It’s a **plug-and-play** model
  - ASR can be retrained and replaced independently
  - Can be simply disabled by passing through the inputs
Part 4:
The long fight with over-suppression
Why it’s a hard problem

- Modern ASR models are already noise-robust
  - Multistyle training (MTR)
  - SpecAugment

- Further masking the input of such a model is risky
  - Early experiments show that adding VoiceFilter-Lite model will:
    - Reduce WER on speech noise cases
    - Increase WER on clean cases and non-speech noise cases

- Most errors in WER are deletion errors – **over-suppression**
Asymmetric loss

- Less tolerant to over-suppression, and more tolerant to under-suppression
  - Conventional reconstruction L2 loss:

\[
L = \sum_{t} \sum_{f} \left( S_{\text{cln}}(t, f) - S_{\text{enh}}(t, f) \right)^2
\]

  - Asymmetric L2 loss, with over-suppression penalty $\alpha > 1$:

\[
g_{\text{asym}}(x, \alpha) = \begin{cases} 
  x & \text{if } x \leq 0 \\
  \alpha \cdot x & \text{if } x > 0
\end{cases}
\]

\[
L_{\text{asym}} = \sum_{t} \sum_{f} \left( g_{\text{asym}} \left( S_{\text{cln}}(t, f) - S_{\text{enh}}(t, f), \alpha \right) \right)^2
\]
Adaptive suppression strength

- We use a weight $w \in [0,1]$ to control suppression strength

$$S^{(t)}_{\text{out}} = w \cdot S^{(t)}_{\text{enh}} + (1 - w) \cdot S^{(t)}_{\text{in}}$$

- We want the weight to be:
  - Larger – when there is overlapped speech (VoiceFilter-Lite is most helpful)
  - Smaller – when the speech is clean, or contains only non-speech noise (VoiceFilter-Lite could be harmful)
Adaptive suppression strength

- We add a side output to the model, and a prediction loss to the total loss.
Adaptive suppression strength

- Denote this noise type prediction as $f_{\text{adapt}}(S_{\text{in}}^{(t)}) \in [0,1]$
  - 0 – clean speech, or containing non-speech noise
  - 1 – overlapped speech

- Adaptive suppression strength:
  $$w^{(t)} = \beta \cdot w^{(t-1)} + (1 - \beta) \cdot (a \cdot f_{\text{adapt}}(S_{\text{in}}^{(t)}) + b)$$
  - $\beta \in [0,1)$ – moving average
  - $a > 0, b \geq 0$ – linear transform
Part 5: Experiment setup
Metrics

- Word Error Rate (WER) is all we need
- Why not signal-to-noise ratio (SNR) or source-to-distortion ratio (SDR)?
  - There is no audio
  - The I/O of VoiceFilter-Lite are ASR features
Models

Speaker recognition (enrollment):

- 3 LSTM layers, each with 768 nodes and 256-dim projection
- 1 final feedforward layer with 256 nodes

VoiceFilter-Lite:

- 3 LSTM layers, each with 512 nodes
- 1 feedforward layer with sigmoid activation for mask prediction (dimension depends on I/O)
- 2 feedforward layers, each with 64 nodes for noise type prediction
Data generation for training and evaluation

- **Clean speech**
- **Room configurations**
  - **Non-speech noise**
    - **Room simulator**
      - **SNR 1dB~10dB**
    - **SNR 1dB~10dB**
  - **Interfering speech**
    - **Room simulator**
      - **Speech noise reverberant**
    - **SNR 1dB~10dB**
  - **Speech additive**
    - **SNR 1dB~10dB**
  - **Non-speech noise**
    - **Non-speech noise additive**
    - **Room simulator**
      - **SNR 1dB~10dB**
  - **Music**
  - **Sound effects**
  - **Cafe ambient noise**
  - **Car ambient noise**
  - **Quiet ambient noise**

**Cases:**
- (case 1): Interfering speech, Clean speech, Non-speech noise
- (case 2): Non-speech noise, Room simulator, SNR 1dB~10dB
- (case 3): Non-speech noise reverberant, Room simulator
- (case 4): Speech additive, SNR 1dB~10dB
- (case 5): Speech noise reverberant, Interfering speech, Room simulator
Two experiment groups

Group 1: LibriSpeech
- RNN-T streaming ASR model trained on LibriSpeech training set
- VoiceFilter-Lite also trained on LibriSpeech
- Evaluate on LibriSpeech testing set (“test-clean” and “test-other”)

Group 2: Realistic speech queries
- RNN-T streaming ASR model trained on YouTube, anonymized voice search, etc.
- VoiceFilter-Lite training and ASR evaluation: Vendor collected dataset of realistic speech queries
- Much more challenging than Group 1:
  - More variations of prosody, sentiment, accent, and acoustic condition
  - Domain mismatch with ASR training set
Part 6:
Results and conclusions
### Results of Group 1 (LibriSpeech)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td>Additive</td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>8.6</td>
<td>35.7</td>
<td>58.5</td>
<td>77.9</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.1</td>
<td>21.5</td>
<td>48.3</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>24.1</td>
<td>50.8</td>
<td>35.5</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.3</td>
<td>23.4</td>
<td>48.9</td>
<td>25.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>24.8</td>
<td>49.8</td>
<td>30.6</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.9</td>
<td>22.2</td>
<td>48.2</td>
<td>23.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>23.9</td>
<td>49.7</td>
<td>30.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w = 0.6$</td>
<td>8.6</td>
<td>24.4</td>
<td>50.7</td>
<td>42.0</td>
</tr>
</tbody>
</table>

- VoiceFilter-Lite consistently improves WER on both non-speech noise and speech noise cases.
## Results of Group 1 (LibriSpeech)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td>Additive</td>
</tr>
<tr>
<td>No voice filtering</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>35.7</td>
<td>58.5</td>
<td>77.9</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>9.1</td>
<td>21.5</td>
<td>48.3</td>
<td>25.5</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>24.1</td>
<td>50.8</td>
<td>35.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>24.8</td>
<td>49.8</td>
<td>30.6</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.3</td>
<td>23.4</td>
<td>48.9</td>
<td>25.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>24.8</td>
<td>49.8</td>
<td>30.6</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.9</td>
<td>22.2</td>
<td>48.2</td>
<td>23.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>23.9</td>
<td>49.7</td>
<td>30.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w = 0.6$</td>
<td>8.6</td>
<td>24.4</td>
<td>50.7</td>
<td>42.0</td>
</tr>
</tbody>
</table>

- L2 loss: WER degrades on clean cases
## Results of Group 1 (LibriSpeech)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td>Additive</td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>8.6</td>
<td>35.7</td>
<td>58.5</td>
<td>77.9</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.1</td>
<td>21.5</td>
<td>48.3</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>24.1</td>
<td>50.8</td>
<td>35.5</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.3</td>
<td>23.4</td>
<td>48.9</td>
<td>25.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>24.8</td>
<td>49.8</td>
<td>30.6</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.9</td>
<td>22.2</td>
<td>48.2</td>
<td>23.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>23.9</td>
<td>49.7</td>
<td>30.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w = 0.6$</td>
<td>8.6</td>
<td>24.4</td>
<td>50.7</td>
<td>42.0</td>
</tr>
</tbody>
</table>

- Asymmetric L2 loss: Less degradation on clean; also less improvement on speech noise cases
## Results of Group 1 (LibriSpeech)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td>Additive</td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>8.6</td>
<td>35.7</td>
<td>58.5</td>
<td>77.9</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.1</td>
<td>21.5</td>
<td>48.3</td>
<td>25.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>24.1</td>
<td>50.8</td>
<td>35.5</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>9.3</td>
<td>23.4</td>
<td>48.9</td>
<td>25.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.6</td>
<td>24.8</td>
<td>49.8</td>
<td>30.6</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>8.9</td>
<td>22.2</td>
<td>48.2</td>
<td>23.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>8.8</td>
<td>23.9</td>
<td>49.7</td>
<td>30.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w = 0.6$</td>
<td>8.6</td>
<td>24.4</td>
<td>50.7</td>
<td>42.0</td>
</tr>
</tbody>
</table>

- Filterbank and stacked filterbank outperform FFT magnitude.
Results of Group 2 (Realistic speech queries)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reverb</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reverb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>15.2</td>
<td>21.1</td>
<td>29.1</td>
<td>56.5</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.4</td>
<td>27.0</td>
<td>36.9</td>
<td>25.1</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.2</td>
<td>22.6</td>
<td>31.2</td>
<td>32.0</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>26.6</td>
<td>35.6</td>
<td>27.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>36.2</td>
<td>36.2</td>
<td>26.8</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.8</td>
<td>25.7</td>
<td>34.4</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 0.3$</td>
<td>15.2</td>
<td>21.7</td>
<td>29.6</td>
<td>42.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$Adaptive , w(t)$</td>
<td>15.3</td>
<td>21.3</td>
<td>29.3</td>
<td>28.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15.4</td>
<td>21.1</td>
<td>29.0</td>
<td>31.4</td>
</tr>
</tbody>
</table>

- L2 and asymmetric L2 loss: WER may degrade on both clean and non-speech noise cases
## Results of Group 2 (Realistic speech queries)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td></td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>15.2</td>
<td>21.1</td>
<td>29.1</td>
<td>56.5</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.4</td>
<td>27.0</td>
<td>36.9</td>
<td>25.1</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.2</td>
<td>22.6</td>
<td>31.2</td>
<td>32.0</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>28.5</td>
<td>38.3</td>
<td>26.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>26.6</td>
<td>35.6</td>
<td>27.5</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>16.7</td>
<td>26.8</td>
<td>36.2</td>
<td>26.8</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.8</td>
<td>25.7</td>
<td>34.4</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 0.3$</td>
<td>15.2</td>
<td>21.7</td>
<td>29.6</td>
<td>42.1</td>
</tr>
<tr>
<td></td>
<td>Adaptive $w(t)$</td>
<td></td>
<td>15.3</td>
<td>21.3</td>
<td>29.3</td>
<td>28.8</td>
</tr>
</tbody>
</table>

- Lower suppression strength ($w = 0.3$) – Tradeoff between:
  - Degradation on clean or non-speech noise
  - Improvement on speech noise
### Results of Group 2 (Realistic speech queries)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Loss</th>
<th>Suppression strength</th>
<th>Clean</th>
<th>Non-speech noise</th>
<th>Speech noise</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additive</td>
<td>Reverb</td>
<td>Additive</td>
</tr>
<tr>
<td>No voice filtering</td>
<td></td>
<td></td>
<td>15.2</td>
<td>21.1</td>
<td>29.1</td>
<td>56.5</td>
</tr>
<tr>
<td>FFT magnitude</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.4</td>
<td>27.0</td>
<td>36.9</td>
<td>25.1</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.2</td>
<td>22.6</td>
<td>31.2</td>
<td>32.0</td>
</tr>
<tr>
<td>Filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>28.5</td>
<td>38.3</td>
<td>26.5</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.3</td>
<td>26.6</td>
<td>35.6</td>
<td>27.5</td>
</tr>
<tr>
<td>Stacked filterbank</td>
<td>L2</td>
<td>$w = 1.0$</td>
<td>16.7</td>
<td>26.8</td>
<td>36.2</td>
<td>26.8</td>
</tr>
<tr>
<td></td>
<td>asym L2, $\alpha = 10$</td>
<td>$w = 1.0$</td>
<td>15.8</td>
<td>25.7</td>
<td>34.4</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$w = 0.3$</td>
<td>15.2</td>
<td>21.7</td>
<td>29.6</td>
<td>42.1</td>
</tr>
<tr>
<td></td>
<td>Adaptive $w(t)$</td>
<td></td>
<td>15.3</td>
<td>21.3</td>
<td>29.3</td>
<td>28.8</td>
</tr>
</tbody>
</table>

- **Adaptive suppression strength:**
  - Minimal degradation on clean or non-speech noise
  - Still big improvement on speech noise
  - Smaller model works similarly well
Results of Group 2 (Realistic speech queries)

- Can the model size go further down?
- No voice filtering baseline:
  - Speech noise, Additive: 56.5%
  - Speech noise, Reverb: 53.8%
Which frontend to use?

- Filterbank and stacked filterbank outperform FFT magnitude
- We prefer stacked filterbank over filterbank
  - More context information
  - Less runtime operations (usually subsample the frames after stacking)
Conclusions

- VoiceFilter-Lite: Use your enrolled voice to improve ASR on overlapped speech
- It’s tiny, fast, streaming, and part of on-device ASR
- Two approaches to resolve over-suppression issues:
  - Asymmetric loss
  - Adaptive suppression strength
- A model of 2.2 MB:
  - No WER degradation on clean and non-speech noise conditions
  - 25.1% (44.4% rel.) WER improvement on overlapped speech
Demo

Android demo for VoiceFilter-Lite and on-device ASR
Acknowledgement

- We’d like to thank Philip Chao, Sinan Akay, John Han, Stephen Wu, Yiteng Huang, Jaclyn Konzelmann and Nino Tasca for the support and helpful discussions.
Questions?