Google

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>

INTERSPEECH 2020

OCTOBER 25-29/ SHANGHAI , CHINA

SHANGHAI INTERNATIONAL CONVENTION CENTER

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

Image from elixirofknowledge.com

Conventional solutions:

- Multi-channel blind separation
- **Single-channel** blind separation

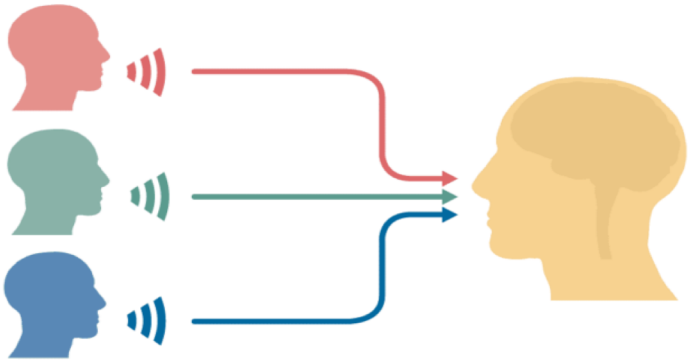
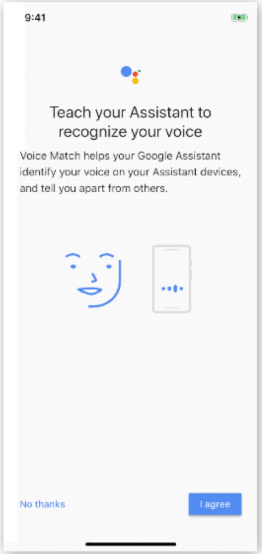
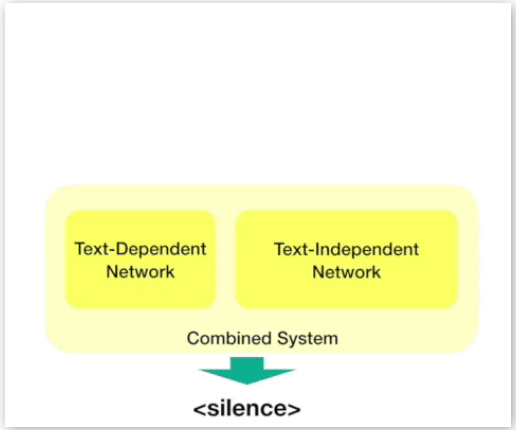


Image from Nima Mesgarani, Columbia University

Knowing “whom to listen to”

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

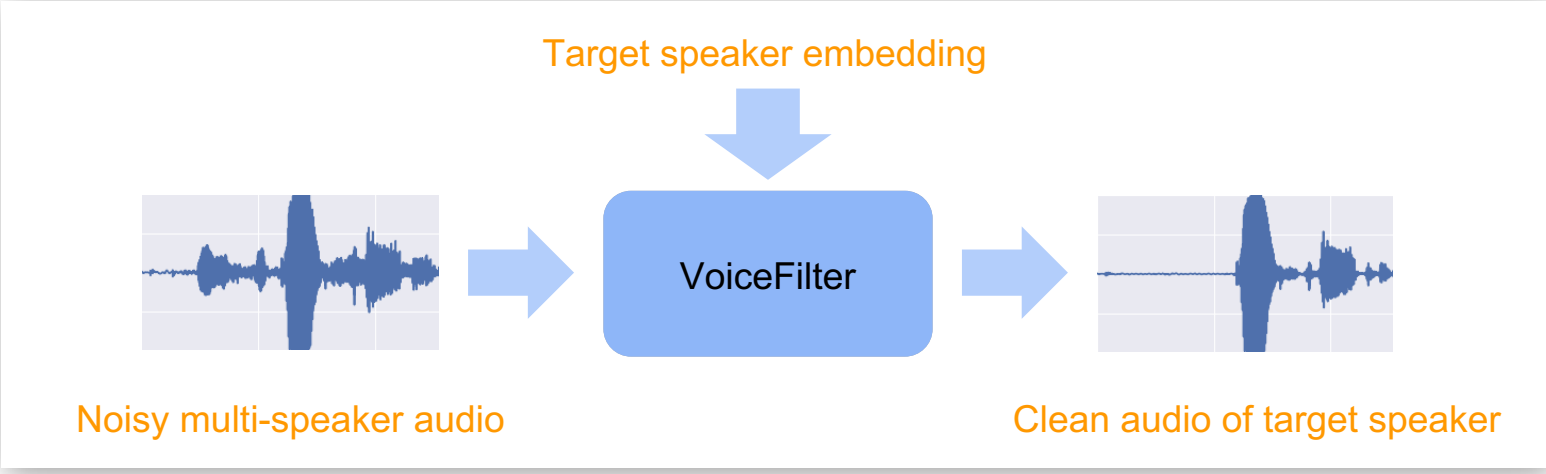




- Blindness sucks
  - Number of sources is unknown
    - Single: don’t want to mess up
    - Multiple: only keep target voice
  - Multiple outputs, still don’t know which one to use (*e.g.* to run ASR)
    - Pick loudest?
    - Run speaker verification on all?
  - Permutation invariance: computational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

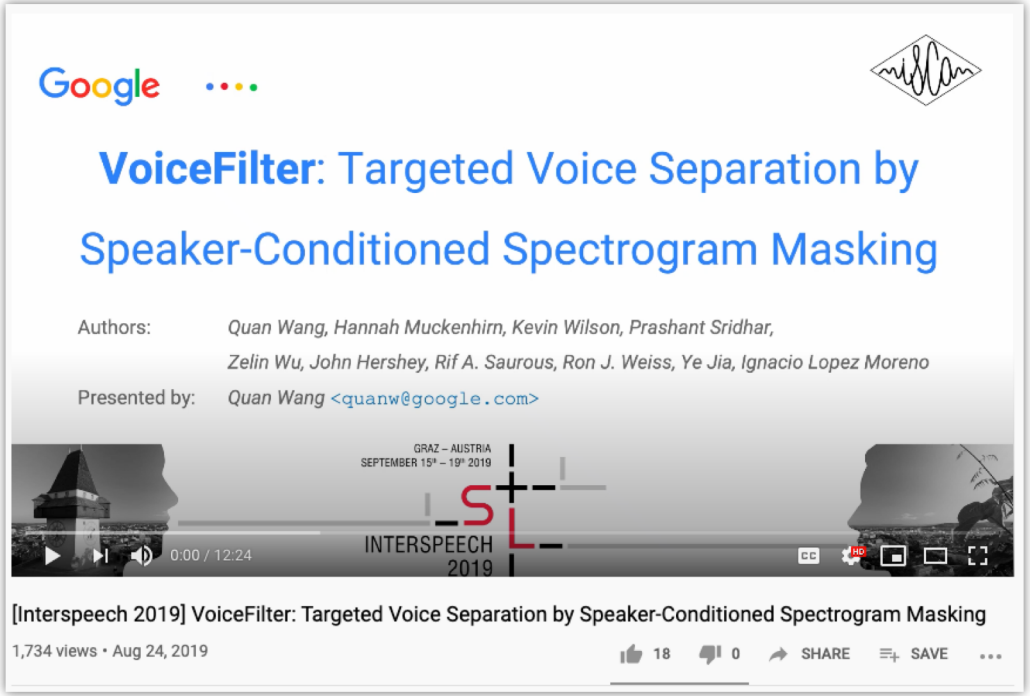
VoiceFilter: Say Goodbye to blindness!

- Condition the voice separation task by target speaker embedding



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

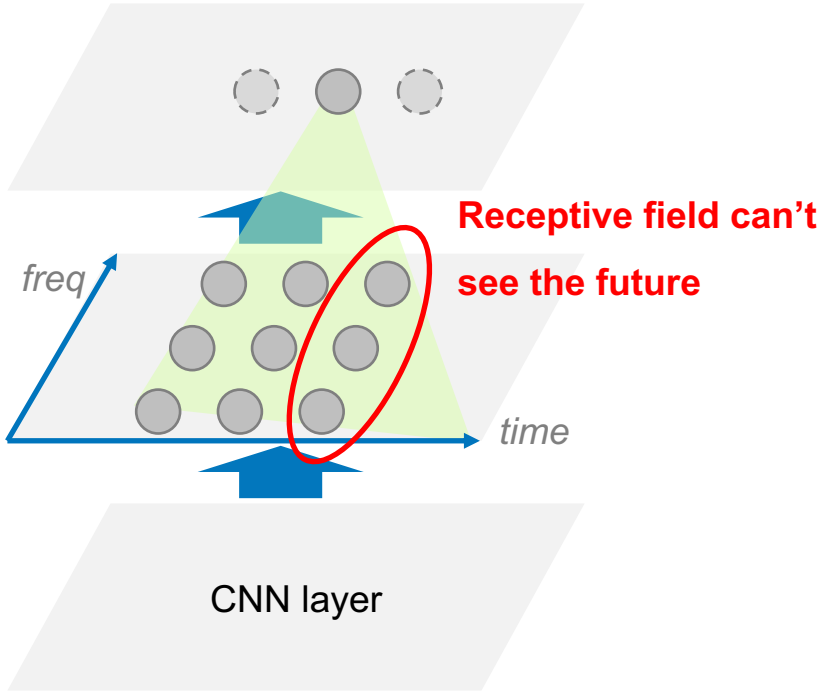
- VoiceFilter: You either always run it, or never run it

Part 3:

The journey to Lite

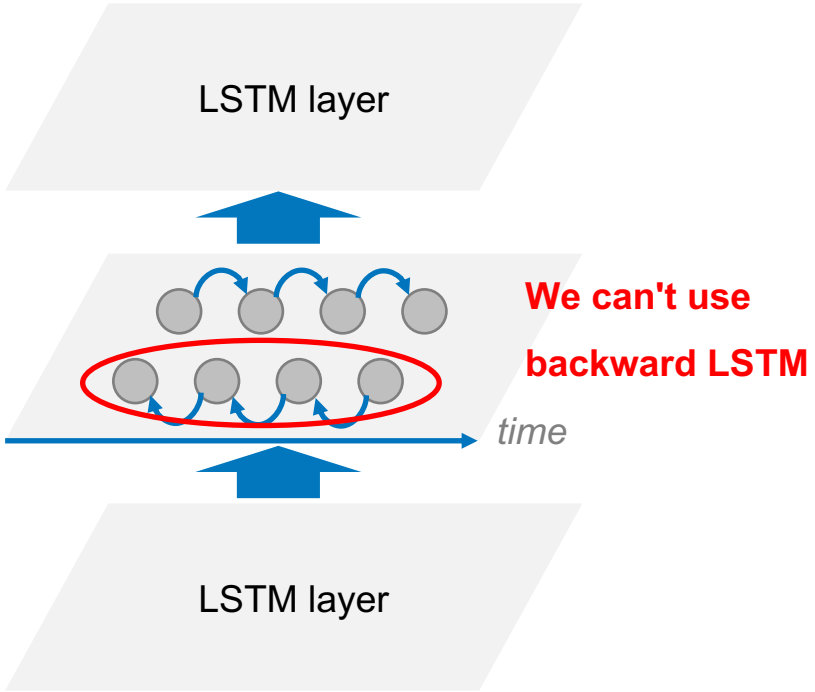
Optimizing for CPU and latency

- Time-frequency CNN
  - Temporal CNN: Not causal
  - Convolutions: Computationally expensive



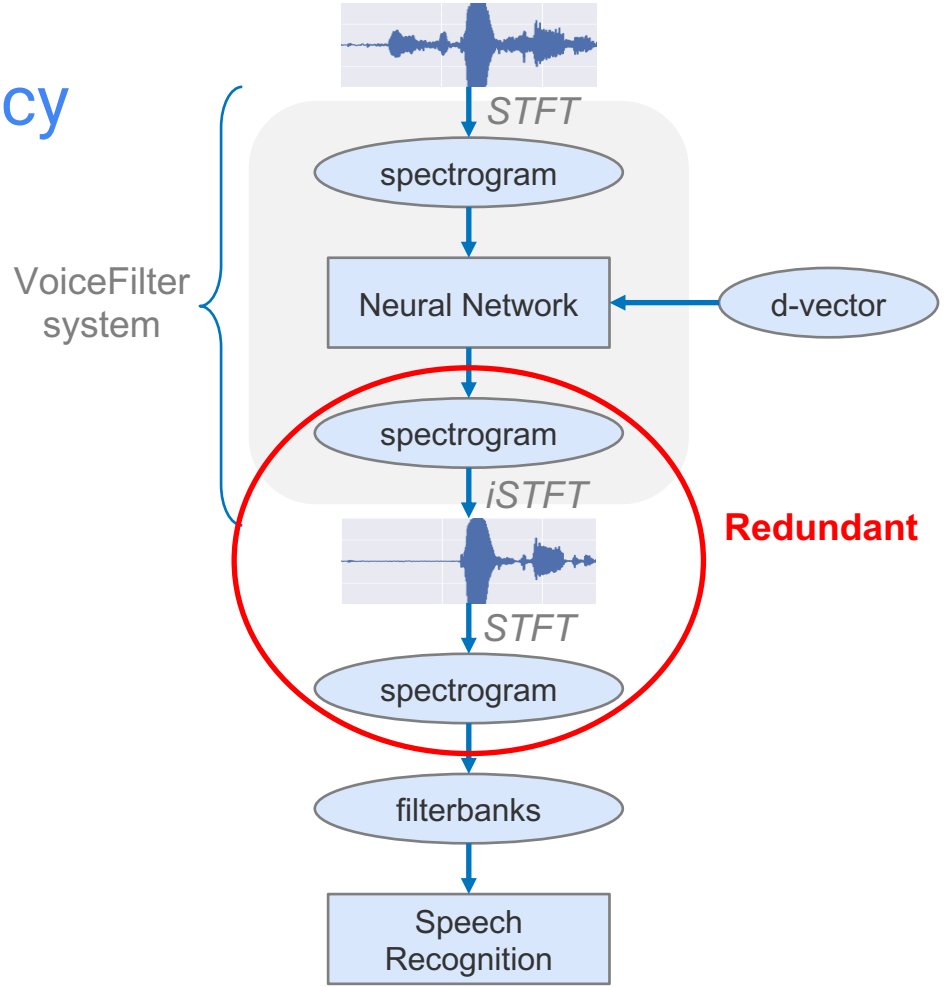
Optimizing for CPU and latency

- Bi-directional LSTM
  - Backward pass: Not causal



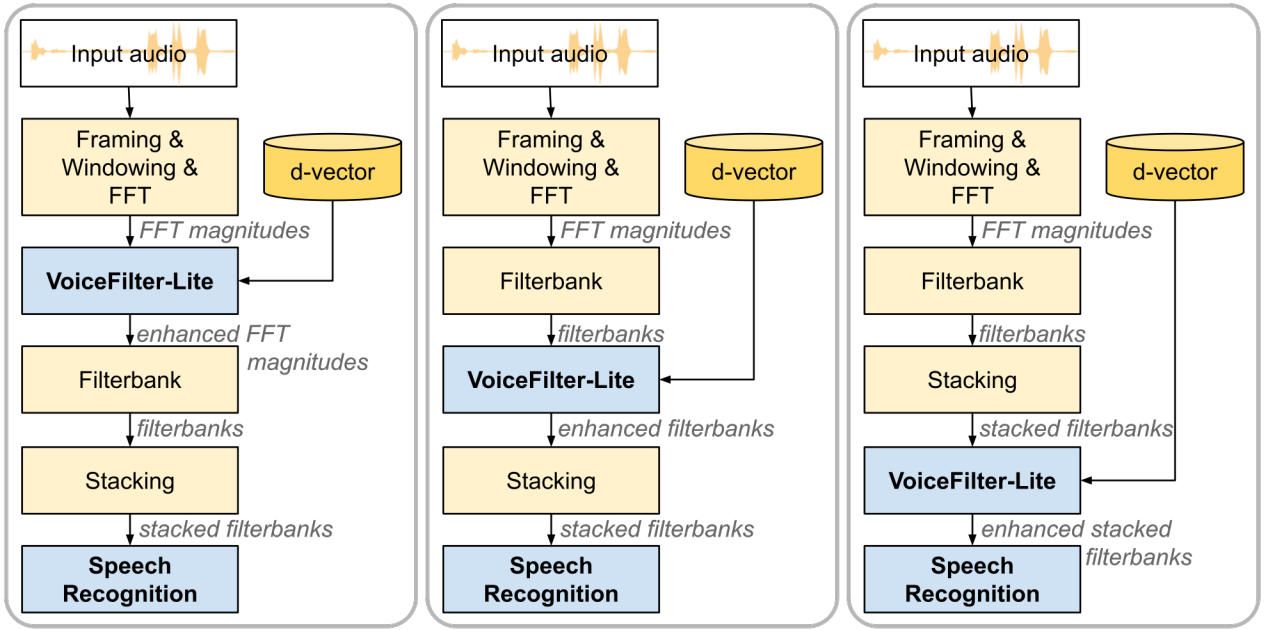
Optimizing for CPU and latency

- 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



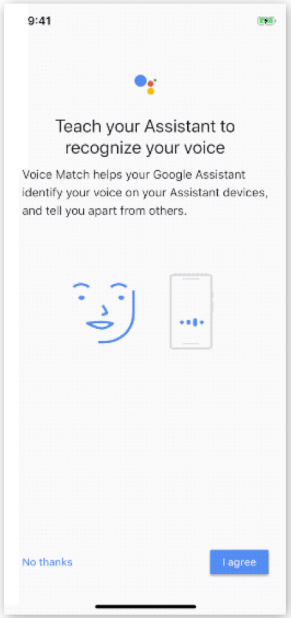
(a) FFT magnitudes as I/O

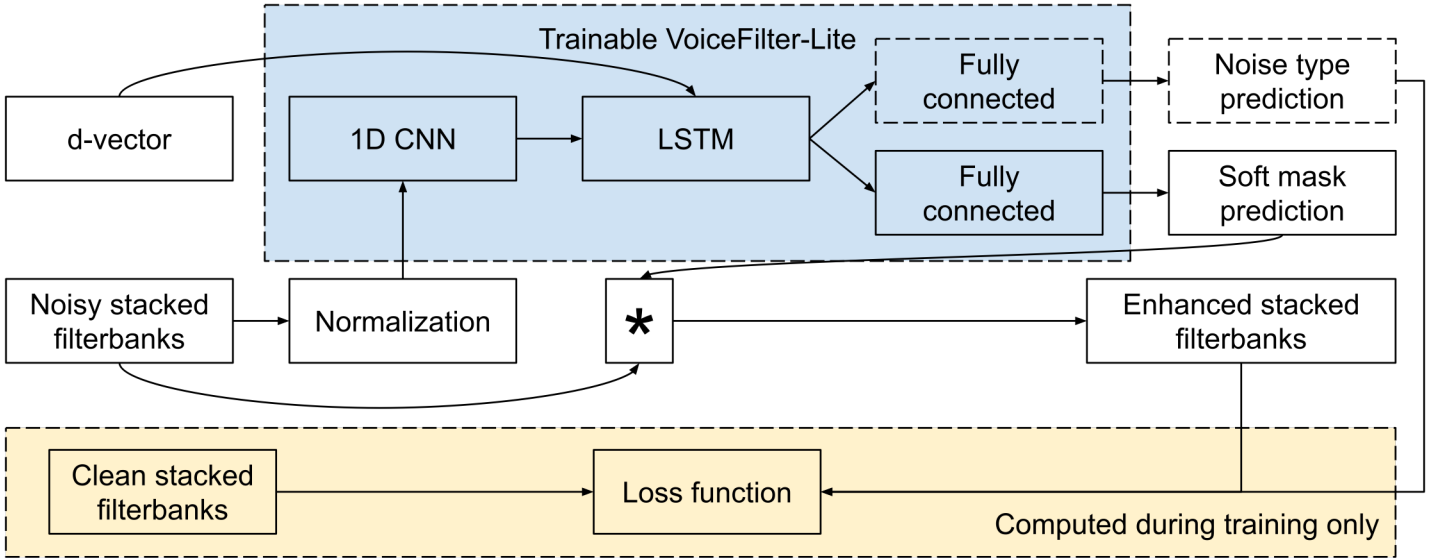
(b) Filterbanks as I/O

(c) Stacked filterbanks as I/O

Model architecture

- Assume we use stacked filterbanks as VoiceFilter-Lite I/O



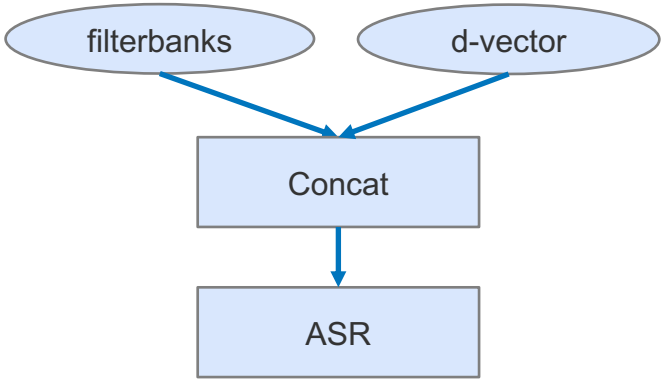


What have changed since VoiceFilter

	VoiceFilter	**VoiceFilter-Lite**
Input & Output	Audio waveform	ASR features (FFT magnitude, filterbank, *etc.*)
Inference	Full sequence offline inference	Online streaming inference (no time-freq CNN, no bi-LSTM)
STFT and iSTFT	Part of VoiceFilter model graph	STFT as part of ASR system; no iSTFT
Model format and size	TensorFlow graph; typically >100MB	Quantized to int8 TFLite model; typically <10MB

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:

clean spectrogram

enhanced spectrogram

$$L = \sum_t \sum_f \left(S_{\text{cln}}(t,f) - S_{\text{enh}}(t,f) \right)^2$$

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

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

$$L_{\text{asym}} = \sum_t \sum_f \left(g_{\text{asym}}\left(S_{\text{cln}}(t,f) - S_{\text{enh}}(t,f), \alpha \right) \right)^2$$

Adaptive suppression strength

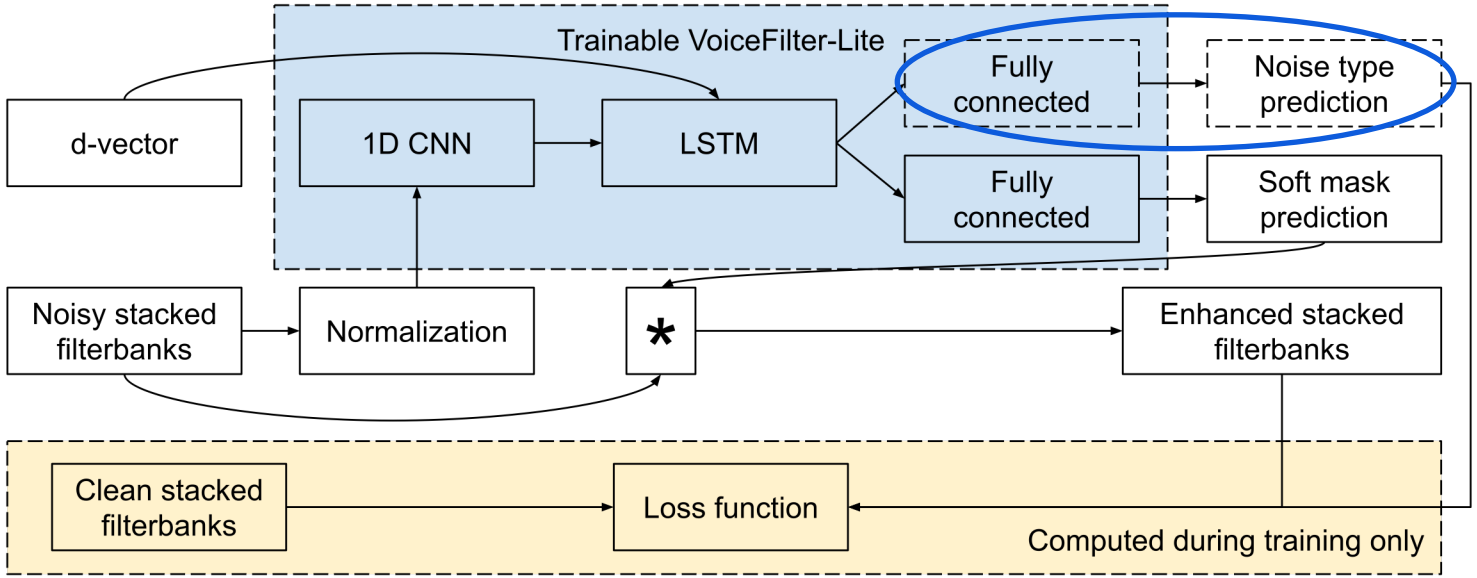
- We use a weight $w \in [0,1]$ to control suppression strength

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

- 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 $f_{\text{adapt}}\left(S_{\text{in}}^{(t)}\right) \in [0,1]$
  - 0 – clean speech, or containing non-speech noise
  - 1 – overlapped speech
- Adaptive suppression strength:

$$w^{(t)} = \beta \cdot w^{(t-1)} + (1-\beta) \cdot \left(a \cdot f_{\text{adapt}}(S_{\text{in}}^{(t)}) + b\right)$$

  - $\beta \in [0,1)$ – 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

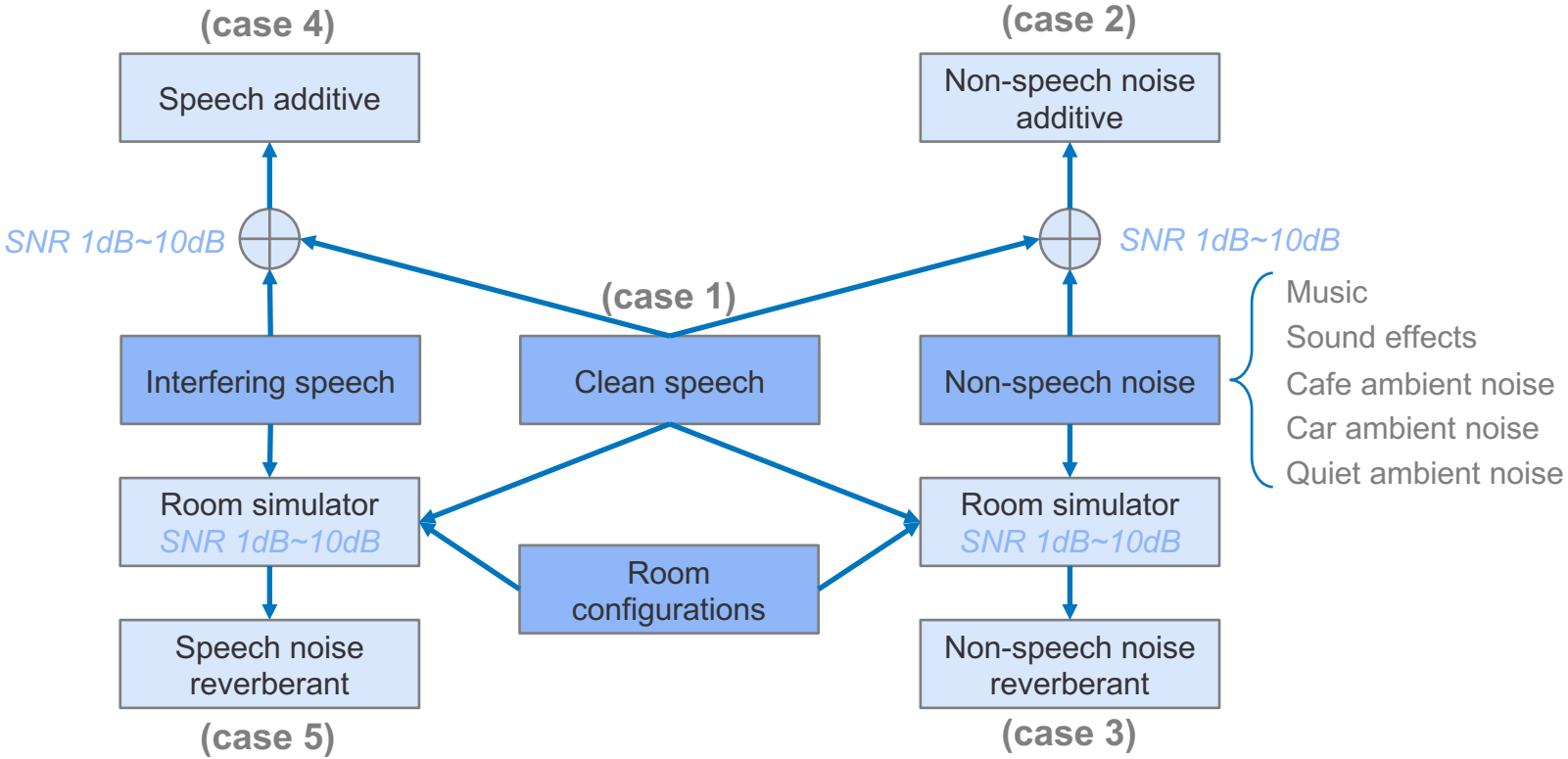
Speaker recognition (enrollment):

- 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



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)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			8.6	35.7	58.5	77.9	79.3	N/A
FFT magnitude	L2	$w = 1.0$	9.1	21.5	48.3	25.5	54.2	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	24.1	50.8	35.5	60.6	
Filterbank	L2	$w = 1.0$	9.3	23.4	48.9	25.4	55.6	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.6	24.8	49.8	30.6	58.4	
Stacked filterbank	L2	$w = 1.0$	8.9	22.2	48.2	23.5	53.7	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	23.9	49.7	30.6	57.8	
		$w = 0.6$	8.6	24.4	50.7	42.0	60.2	

- VoiceFilter-Lite consistently improves WER on both non-speech noise and speech noise cases

Results of Group 1 (LibriSpeech)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			8.6	35.7	58.5	77.9	79.3	N/A
FFT magnitude	L2	$w = 1.0$	9.1	21.5	48.3	25.5	54.2	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	24.1	50.8	35.5	60.6	
Filterbank	L2	$w = 1.0$	9.3	23.4	48.9	25.4	55.6	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.6	24.8	49.8	30.6	58.4	
Stacked filterbank	L2	$w = 1.0$	8.9	22.2	48.2	23.5	53.7	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	23.9	49.7	30.6	57.8	
		$w = 0.6$	8.6	24.4	50.7	42.0	60.2	

- L2 loss: WER degrades on clean cases

Results of Group 1 (LibriSpeech)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			8.6	35.7	58.5	77.9	79.3	N/A
FFT magnitude	L2	$w = 1.0$	9.1	21.5	48.3	25.5	54.2	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	24.1	50.8	35.5	60.6	
Filterbank	L2	$w = 1.0$	9.3	23.4	48.9	25.4	55.6	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.6	24.8	49.8	30.6	58.4	
Stacked filterbank	L2	$w = 1.0$	8.9	22.2	48.2	23.5	53.7	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	23.9	49.7	30.6	57.8	
		$w = 0.6$	8.6	24.4	50.7	42.0	60.2	

- Asymmetric L2 loss: Less degradation on clean; also less improvement on speech noise cases

Results of Group 1 (LibriSpeech)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			8.6	35.7	58.5	77.9	79.3	N/A
FFT magnitude	L2	$w = 1.0$	9.1	21.5	48.3	25.5	54.2	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	24.1	50.8	35.5	60.6	
Filterbank	L2	$w = 1.0$	9.3	23.4	48.9	25.4	55.6	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.6	24.8	49.8	30.6	58.4	
Stacked filterbank	L2	$w = 1.0$	8.9	22.2	48.2	23.5	53.7	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	8.8	23.9	49.7	30.6	57.8	
		$w = 0.6$	8.6	24.4	50.7	42.0	60.2	

- Filterbank and stacked filterbank outperform FFT magnitude

Results of Group 2 (Realistic speech queries)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			15.2	21.1	29.1	56.5	53.8	N/A
FFT magnitude	L2	$w = 1.0$	15.4	27.0	36.9	25.1	36.8	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.2	22.6	31.2	32.0	37.8	
Filterbank	L2	$w = 1.0$	15.3	28.5	38.3	26.5	38.5	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.3	26.6	35.6	27.5	37.4	
Stacked filterbank	L2	$w = 1.0$	16.7	26.8	36.2	26.8	37.4	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.8	25.7	34.4	27.4	36.7	
		$w = 0.3$	15.2	21.7	29.6	42.1	43.6	
		Adaptive $w^{(t)}$	15.3	21.3	29.3	28.8	37.2	
			15.4	21.1	29.0	31.4	39.1	2.2 MB

- L2 and asymmetric L2 loss: WER may degrade on both clean and non-speech noise cases

Results of Group 2 (Realistic speech queries)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			15.2	21.1	29.1	56.5	53.8	N/A
FFT magnitude	L2	$w = 1.0$	15.4	27.0	36.9	25.1	36.8	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.2	22.6	31.2	32.0	37.8	
Filterbank	L2	$w = 1.0$	15.3	28.5	38.3	26.5	38.5	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.3	26.6	35.6	27.5	37.4	
Stacked filterbank	L2	$w = 1.0$	16.7	26.8	36.2	26.8	37.4	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.8	25.7	34.4	27.4	36.7	
		$w = 0.3$	15.2	21.7	29.6	42.1	43.6	
		Adaptive $w^{(t)}$	15.3	21.3	29.3	28.8	37.2	
			15.4	21.1	29.0	31.4	39.1	2.2 MB

- Lower suppression strength ($w = 0.3$) – Tradeoff between:
  - Degradation on clean or non-speech noise
  - Improvement on speech noise

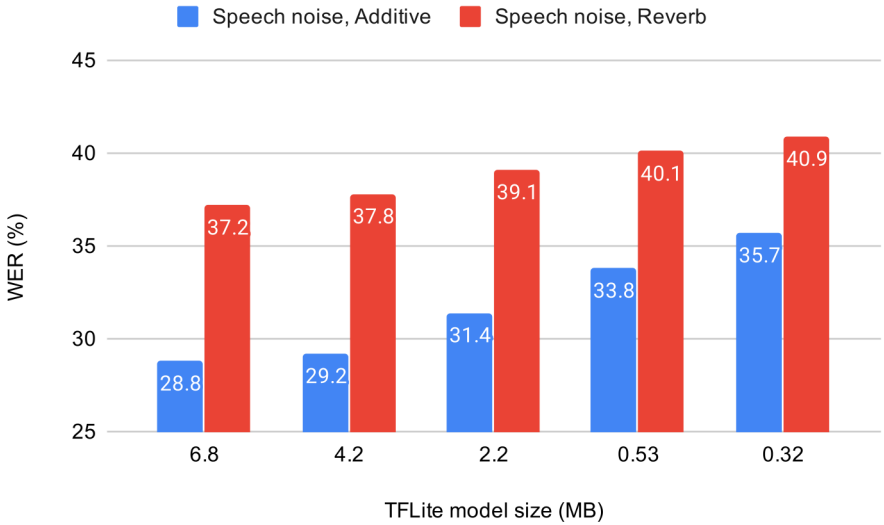
Results of Group 2 (Realistic speech queries)

Feature	Loss	Suppression strength	Clean	Non-speech noise		Speech noise		Size
				Additive	Reverb	Additive	Reverb	
No voice filtering			15.2	21.1	29.1	56.5	53.8	N/A
FFT magnitude	L2	$w = 1.0$	15.4	27.0	36.9	25.1	36.8	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.2	22.6	31.2	32.0	37.8	
Filterbank	L2	$w = 1.0$	15.3	28.5	38.3	26.5	38.5	5.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.3	26.6	35.6	27.5	37.4	
Stacked filterbank	L2	$w = 1.0$	16.7	26.8	36.2	26.8	37.4	6.8 MB
	asym L2, $\alpha = 10$	$w = 1.0$	15.8	25.7	34.4	27.4	36.7	
		$w = 0.3$	15.2	21.7	29.6	42.1	43.6	
		Adaptive $w^{(t)}$	15.3	21.3	29.3	28.8	37.2	
			15.4	21.1	29.0	31.4	39.1	2.2 MB

- 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%



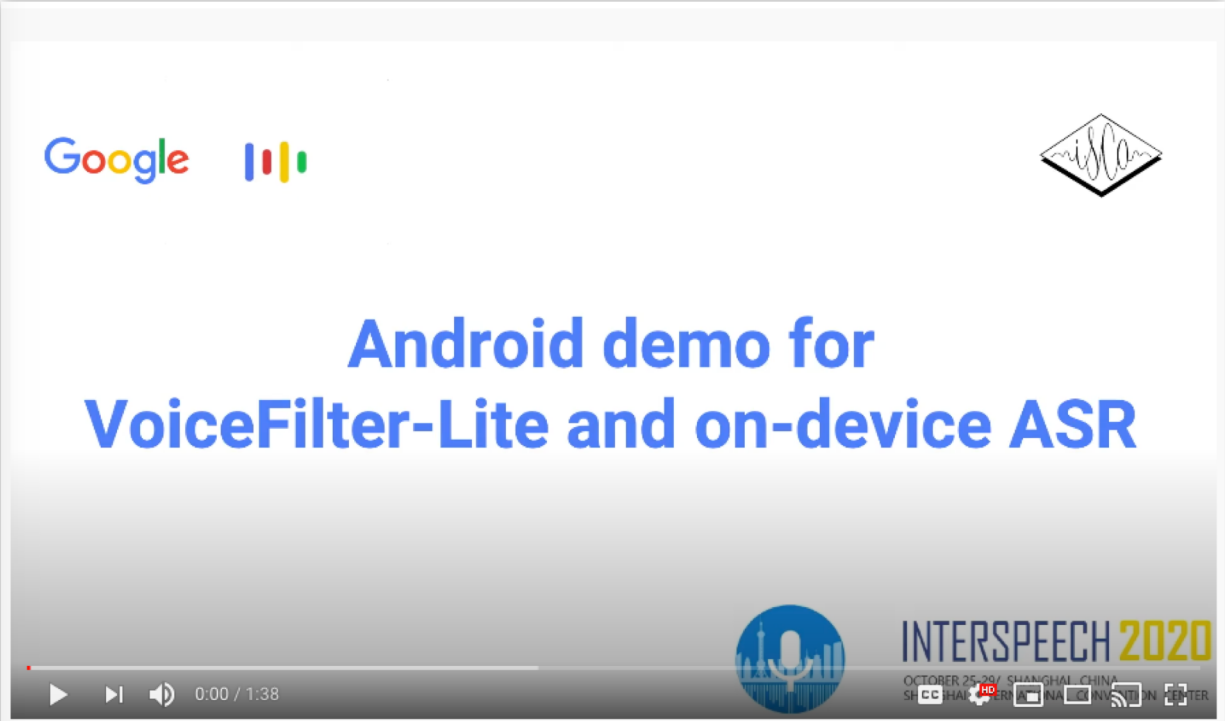
Which frontend to use?

- 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?