

Algorithms and Evaluation Guide v1.6

FemtoseNSE includes the pre-trained algorithms below alongside its evaluation kit. This guide details the algorithms and suggests a process for testing them.

Type	Model Name	Description	Support ¹
Always-on Keyword Detection	WWDALEXA_8khz_16ms_v0	Wake Word Detection “Alexa” 1 microphone 8 kHz sampling rate, 16 ms hop-size Version 0	EVK2 EVK2v2
	GSC_8khz_16ms_v0	Keyword Detection (Google Speech Commands) 1 microphone 8 kHz sampling rate, 16 ms hop-size Version 0	EVK2 EVK2v2 EVK3
AI Noise Reduction	AINRGP_16khz_4hop_8algo_v4	AI Noise Reduction “General Purpose” 1 microphone 16 kHz sampling rate, 4 ms hop-size 8 ms algorithmic latency Version 4	EVK2 EVK2v2 EVK3
	AINRGP_16khz_1hop_2algo_v2 ²	AI Noise Reduction “General Purpose” 1 microphone 16 kHz sampling rate, 1 ms hop-size 2 ms algorithmic latency Version 2	EVK2 EVK2v2
	AINRGP_16khz_1hop_2algo_v3 ³	AI Noise Reduction “General Purpose” 1 microphone 16 kHz sampling rate, 1 ms hop-size 2 ms algorithmic latency Version 3	EVK2v2
Spoken Language Understanding	EN-SLU_SH_8khz_16ms_v0	English SLU “Smart Home” 1 microphone 8 kHz sampling rate, 16 ms hop-size Version 0	EVK2 EVK2v2
	KR-SLU_SH_8khz_16ms_v0	Korean SLU “Smart Home” 1 microphone 8 kHz sampling rate, 16 ms hop-size Version 0	EVK2 EVK2v2

¹ EVK2 was released with SPU-001 Test Chip 2 (SPU-001-TC2), and EVK2v2 was released with the SPU-001 mass production chip (SPU-001-MP). Some models are supported by EVK2v2 but not EVK2. However, EVK2 can be easily upgraded to EVK2v2 with an updated evaluation board (EVB) and PCB connector and updating firmware (See Quick Start Guide, Application Note 006). Please reach out to your Femtosense contact if you have EVK2 and would like to request parts for updating your EVK, free of charge.

² Note that AINRGP_16khz_1hop_2algo_v2 will be referred to as ClaraMono-2ms v1.0 in future releases and later in this document

³ Note that AINRGP_16khz_1hop_2algo_v3 will be referred to as ClaraMono-2ms v1.5 in future releases and later in this document

Table of Contents

1. Algorithms.....3

1.1 Wake Word Detection Algorithms.....3

1.2 Keyword Detection.....4

1.3 AI Noise Reduction Algorithms.....5

1.4 Spoken Language Understanding Algorithms.....6

2. Proposed Evaluation.....8

2.1 Wake Word Detection.....8

2.2 Multi-Class Keyword Detection (Google Speech Commands).....9

2.3 AI Noise Reduction.....9

2.3.1 8ms Latency Model.....11

2.3.2 2ms Latency Models (comparison between v2 and v3).....12

2.4 Spoken Language Understanding.....15

3. Appendix.....16

3.1 Appendix A.....16

4. Change Log.....17

1. Algorithms

1.1 Wake Word Detection Algorithms

Wakeword detection (WWD) algorithms recognize the single words or phrases like “Alexa”, “Hey Siri”, “Ok Google”, and “Hi Bixby.” The model input is a sequence of raw waveform frames, and its output is a sequence of probabilities of the presence of the keyword at each frame. The reference models below have been trained against indoor environmental noise, competing speech, and room reverberance.

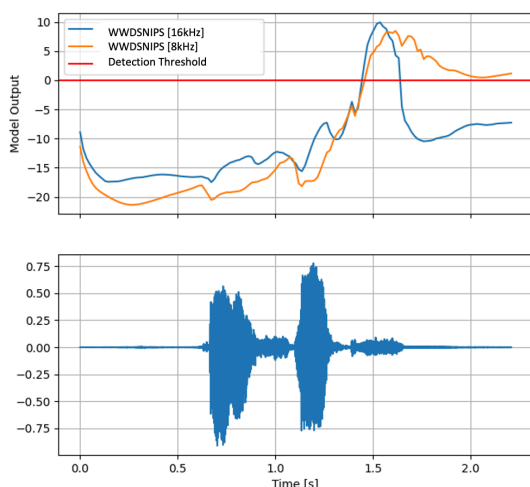


Figure 1: Wakeword detection algorithms return scalar values for each input frame of audio (every 16ms). When the output exceeds the detection threshold (default 0), it is interpreted as a positive detection of the keyword.

Variants exist for 16kHz and 8kHz sampling rates.

Model Naming Convention Example

- Target keyword: “Hey Snips” or “Alexa”
- Input audio sampling rate: 8kHz
- Hop size: 16ms
- Algorithm version: 0



Summary of Models

Model Name	Compiler/ HW Version	VDD+VDDM Power Consumption (Optimized EVK2,EVK2v2)	Execution Time (Optimized EVK2,EVK2v2)	VDD+VDDM Power Consumption (Unoptimized EVK3 ⁴)
WWDALEXA_8khz_16ms_v0	0.2.8 TC2 chip	135μW @ 22°C	15.3ms	N/A
WWDALEXA_8khz_16ms_v0	0.5.0 MP chip	165μW @ 21°C	14.2ms	N/A

⁴EVK3 does not have firmware-based power optimizations as of the current release, EVK3 v1.0.3. Power optimized models should be measured on EVK2. Power optimizations include sleep/wake cycling, tuning clock frequency, and more.

1.2 Keyword Detection

Keyword detection (KWS) algorithms recognize one of several keywords or phrases. The model input is a sequence of raw waveform frames, and its output is a sequence of probabilities of the presence of each keyword at each frame. The reference models below were trained on the [Google Speech Commands](#) (GSC) dataset, recognizing 11 specified keywords: ["On", "Off", "Left", "Right", "Up", "Down", "Yes", "No", "Stop", "Go", "Hey Snips"⁵].

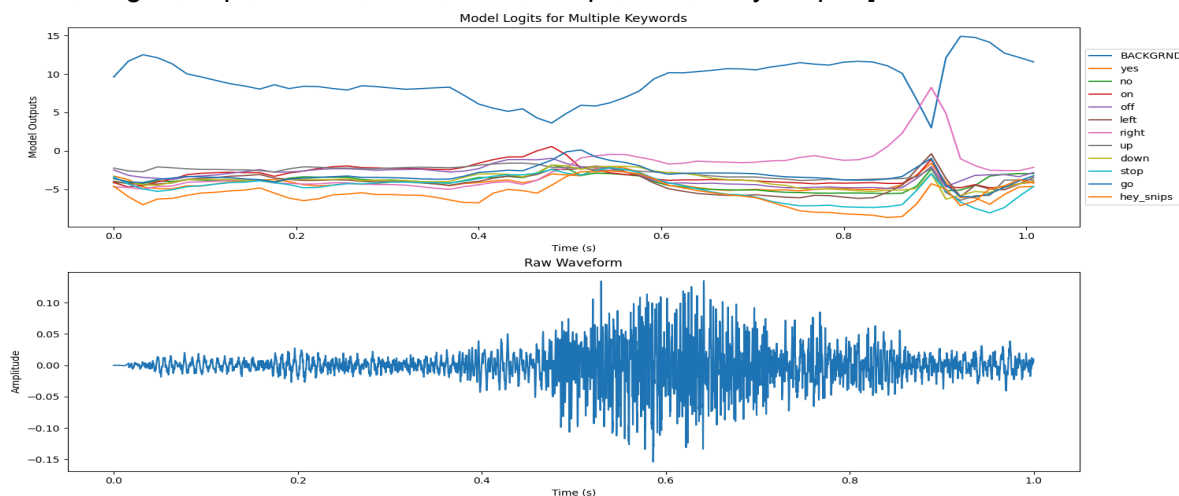
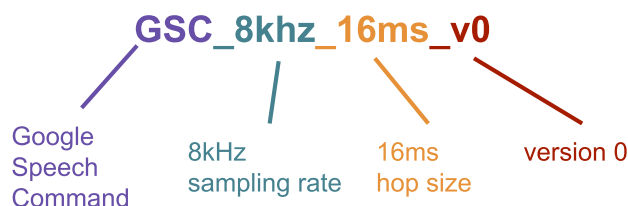


Figure 2: Keyword Detection (class: "right") example using GSC_8Khz_16ms_v0 model. Model outputs represent confidence scores for each keyword class, with peaks indicating likely keyword detection. "BACKGRND" represents the background class.

To improve robustness, other keywords from the GSC dataset serve as "distractors," countering potential false activations. The models have been trained in conditions with indoor environmental noise, competing speech, and room reverberance, and processes audio input in the INT16 PCM format at a sampling rate of 8 KHz.

Model Naming Convention Example

- Target keywords: "On", "Off", "Left", "Right", "Up", "Down", "Yes", "No", "Stop", "Go", "Hey Snips"
- Input audio sampling rate: 8kHz
- Hop size: 16ms
- Algorithm version: 0



⁵ "Hey Snips" was incorporated from a separate dataset

Summary of Models

Model Name	Compiler/ HW Version	VDD+VDDM Power Consumption (Optimized EVK2,EVK2v2)	Execution Time (Optimized EVK2,EVK2v2)	VDD Power Consumption (Unoptimized EVK3')
GSC_8khz_16ms_v0	0.2.8 TC2 chip	400 μ W @ 22°C	15.7ms	740 μ W
GSC_8khz_16ms_v0	0.5.0 MP chip	403 μ W @ 21°C	15.1ms	N/A

1.3 AI Noise Reduction Algorithms

AI Noise Reduction (AINR) algorithms remove background noise while preserving human speech. Speech can be in any language. The model's input is a sequence of noisy raw waveform frames, and its output is a sequence of enhanced waveform frames. Both input and output audio use an INT16 PCM format.

Model Naming Convention Example

- AINR type: General Purpose
- Input/output audio sampling rate: 16kHz
- Hop size: 4ms
- Algorithmic latency: 8ms
- Algorithm version: 4



Summary of models

Model Name	Algo Latency	Compiler/HW Version	VDD+VDDM Power Consumption (Optimized EVK2,EVK2v2)	Execution Time (Optimized EVK2,EVK2v2)	VDD Power Consumption (Unoptimized EVK3')
AINRGP_16khz_4hop_8algo_v4	8 ms	0.2.8 TC2 chip	960 μ W @ 22°C	1ms	1.3mW
AINRGP_16khz_4hop_8algo_v4	8 ms	0.5.0 MP chip	809 μ W @ 22°C	1ms	N/A
AINRGP_16khz_1hop_2algo_v2	2 ms	0.5.0 TC2 chip	4.85mW @ 22°C	890 μ s	N/A
AINRGP_16khz_1hop_2algo_v2	2 ms	0.5.0 MP chip	3.42mW @ 23°C	720 μ s	N/A
AINRGP_16khz_1hop_2algo_v3	2 ms	0.5.0 MP chip	4.24mW @ 23°C	828 μ s	N/A

1.4 Spoken Language Understanding Algorithms

Spoken Language Understanding (SLU) algorithms recognize a speaker's intention from voice commands across a variety of formulations. For example, saying "Volume up", "Increase the volume", or "Make it louder" to a smart speaker should all result in it having a higher volume. Model inputs are a sequence of raw waveform frames. Model outputs are a sequence of probabilities across the intent categories. We provide two reference models for intents spoken in English and Korean respectively.

English Smart Home SLU

Intent/Action	English Phrases
Control Lights	<ul style="list-style-type: none"> • Lights [on/off] • [Kitchen/Bedroom/Washroom] Lights [on/off] • [Switch/Turn] [on/off] (the) lights • [Switch/Turn] [on/off] (the) lights in the [kitchen/ bedroom/washroom] • [Switch/Turn] (the) lights [on/off] • [Switch/Turn] (the) lights [on/off] in the [kitchen/bedroom/washroom]
Control Lamp	<ul style="list-style-type: none"> • Lamp [on/off] • [Switch/Turn] [on/off] (the) lamp • [Switch/Turn] (the) lamp [on/off]
Control Music	<ul style="list-style-type: none"> • Turn [on/off] (the) music. • [Play/Resume] (the) music • [Stop/Pause] (the) music
Control TV Language	<ul style="list-style-type: none"> • [Change/Set/Switch] the language to [Chinese/English/German/Korean] • Allow a different language • Set my TV's language to [Chinese/English/German/Korean] • I need to practice my [Chinese/English/German/Korean], Switch the language
Control Volume	<ul style="list-style-type: none"> • Volume [up/down] • Volume [max/mute] • Louder/Quieter • [Increase/Decrease] (the) [volume/sound/sound volume] • Turn (the) [volume/sound] [up/down] • Turn [up/down] (the) [volume/sound] • Turn it [up/down] • Far too [quiet/loud] • Make [it/the music] [louder/quieter]

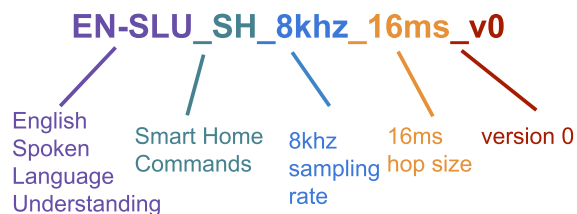
Each Intent/Action on the left can be executed with the via speaking the phrases on the right. Words in [square brackets] are various options, and words in (parenthesis) are optional and may be omitted.

Korean Smart Home SLU

Intent/Action	Korean Phrases ⁶
Control Lights	<ul style="list-style-type: none"> ● 불 [켜줘/꺼줘]. ● 조명을 [켜/꺼] 주세요. ● 불을 [키/끄]는 게 어때요? ● [화장실/침실/부엌] 불 [켜줘/꺼줘]. ● [화장실/침실/부엌] 조명 [키/끄] 는 걸 도와줄 수 있어?
Control Music	<ul style="list-style-type: none"> ● 음악 [켜/꺼]. ● 음악 플레이어를 [활성화해/비활성화해].
Control TV	<ul style="list-style-type: none"> ● 언어를 [영어로/독일어로/한국어로/중국어로] [설정해/전환해/바꾸고 싶어요]. ● TV [켜줘/꺼줘]. ● TV [켜도/꺼도] 되나요?
Control Volume	<ul style="list-style-type: none"> ● 소리 [올려/낮춰]. ● 소리 더 [높이고/낮추고] 싶어. ● 더 [크게 부탁해/조용하게 부탁할게].

Model Naming Convention Example

- SLU Language: English, Korean
- Command purpose: Smart Home
- Input/output audio sampling rate: 8 kHz
- Hop size: 16ms
- Algorithm version: 0



Model Name	Compiler/HW Version	VDD+VDDM Power Consumption (Optimized EVK2, EVK2v2)	Execution Time (Optimized EVK2, EVK2v2)	VDD+VDDM Power Consumption (Unoptimized EVK3 ⁷)
EN-SLU_SH_8kHz_16ms_v0	0.2.8 TC2 chip	436μW @ 23°C	15.1ms	N/A
EN-SLU_SH_8kHz_16ms_v0	0.5.0 MP chip	420μW @ 22°C	15.2ms	N/A
KR-SLU_SH_8kHz_16ms_v0	0.2.8 TC2 chip	447μW @ 23°C	15.1ms	N/A
KR-SLU_SH_8kHz_16ms_v0	0.5.0 MP chip	419μW @ 23°C	14.9ms	N/A

⁶See full list of command in [Appendix A](#)

⁷EVK3 does not have firmware-based power optimizations as of the current release, EVK3 v1.0.3. Power optimized models should be measured on EVK2. Example power optimizations include sleep/wake cycling, tuning clock frequency, and shutting off unused cores.

2. Proposed Evaluation

2.1 Wake Word Detection

Our algorithm detects `Alexa` in a large variety of environments. We recommend evaluating the algorithm in the following conditions:

- Noise Environments: music noise, competing speech
- Signal to Noise Ratios: 0dB SNR or higher
- Distance: Play source audio at a distance from 0 to 2 meters from the microphone.

The model was trained on a small dataset of speakers with American accents. Performance may degrade for speakers with other accents. To aid in the evaluation, we provide a small set of validation audio files for testing purposes. These audio samples were not used during the training of the model. Testing should be conducted in a non-reverberant environment when mixing with noise, otherwise the effective SNRs levels will be lower.

Specifications:

- **Audio In:**
 - 8 kHz Sampling Rate
 - Monaural
 - 16 bits (pcm)
- **Output:** scalar probability of keyword

Model Performance:

We measured the performance of our algorithm in TV noise across different SNR levels (0dB, +3dB, +6dB, +9dB, +20dB), as well as in silence at a distance (1 meter, 2 meters, 4 meters, 8 meters and 16 meters). We use the True Positive Rate (TPR) to measure the ability of our algorithm to detect a keyword in noisy and far-field conditions. **We measured 0 False Positives events over the course of 56 hours of TV noise.**

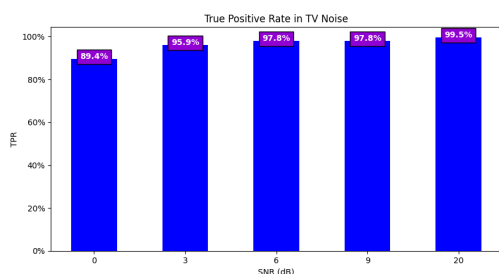


Figure 3: True Positive Rate (TPR) for the WWDALLEXA_8khz_16ms_v0 on a test set of 3,605 samples played with TV background noise from the [TVSM-cuesheet Dataset](#).

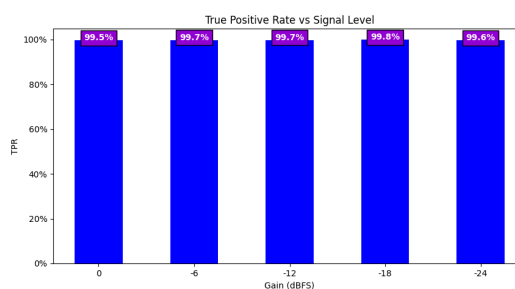


Figure 4: True Positive Rate (TPR) for the WWDALLEXA_8khz_16ms_v0 on a test set of 3,605 samples played in silence at different distances. The reference at 0 dBFS corresponds to 1m distance of the speaker and device, and each added -6dBFS corresponds to doubling the distance between speaker and receiver microphone.

2.2 Multi-Class Keyword Detection (Google Speech Commands)

The model recognizes a set of predefined keywords: "On", "Off", "Left", "Right", "Up", "Down", "Yes", "No", "Stop", "Go", and "Hey Snips" in a variety of environments. We recommend evaluating the algorithm in the following conditions:

- Noise Environments: music noise, competing speech
- Signal to Noise Ratios: 3dB SNR or higher
- Distance: Play source audio at a distance from 0 to 2 meters from the microphone.

The model was trained using the Google Speech Commands dataset, with additional "Hey Snips" utterances from another dataset. To recognize various accents, including English, Korean, Malaysian, Singaporean, and Indonesian, it was also trained with extra data from a third-party provider specializing in foreign accents.

Specifications:

- **Audio In:**
 - 8 kHz Sampling Rate
 - Monaural
 - 16 bits (pcm)
- **Output:** 12 confidence scores corresponding to each Keyword and Background class

Model Performance:

The model has been rigorously tested under challenging audio conditions with files randomly mixed at a Signal-to-Noise Ratio (SNR) ranging between 3-9 dB. The target keywords and distractors were taken from the Eval-Set of the [Google Speech Commands Dataset](#). The "Hey Snips" samples were sourced from a separate dataset. Background noise samples were incorporated from various datasets including [WHAM!](#), [Epic-Kitchens](#), [DNS Challenge](#), and [Vehicle Interior Sound](#) datasets to simulate real-world scenarios. The evaluation includes precision and recall metrics, averaged both across all keywords and specifically for the background class, ensuring a comprehensive analysis of its accuracy and reliability in different scenarios, with separate metrics provided for reverb and non-reverb cases.

Metric Type	Average Across Keywords (%)	Background Class (%)
Precision	88.35	95.63
Recall	89.36	96.15

Table 1: GSC Eval Metrics in Non-Reverb Environment for GSC_8khz_16ms_v0 model

Metric Type	Average Across Keywords (%)	Background Class (%)
Precision	82.32	94.88
Recall	85.89	93.95

Table 2: GSC Eval Metrics in Reverberant Environment for GSC_8khz_16ms_v0 model

2.3 AI Noise Reduction

Our AINR algorithms remove background noise while preserving the speech. We recommend evaluating the algorithms in the following conditions:

- Noise Environments: car, **babble (restaurant/café background)**, colored noise and transient sounds
- Signal to Noise Ratios: The algorithm should provide good performance across a range of different SNR conditions.
- Distance: Play source audio at a distance from 0 to 3 meters from the microphone.

We suggest experiencing the algorithm while wearing an Active Noise Cancellation (ANC) headset to reduce noise and speech that reaches the user's ears directly through the physical earbud or headset enclosure. This approach replicates the usage scenario where our algorithm would be combined with ANC in earbuds to mitigate the direct path. For assistive hearing devices, patients with hearing loss experience less of a direct path than people with normal hearing so ANC may not be needed.

Specifications:

- Audio In:
 - 16 kHz Sampling Rate
 - Monaural
 - 16 bits (PCM)
- Audio out:
 - 16 kHz Sampling Rate
 - Monaural
 - 16 bits (PCM)

Our algorithm is not trained for a specific microphone model. For reference, the microphone we used for testing had the following specifications.

- MEMS microphone
- SNR: 67.5 dB
- AOP: 123 dB
- Sensitivity: -38 ± 3 dBFS

Model Performance:

The model works well across a wide range of SNR conditions. At very low SNRs, there may be more noise transparency and the algorithm may distort the speech.

We measured the performance of our algorithm with six different metrics across SNR levels (-6dB, -3dB, 0dB, +3dB, +6dB) for car and babble speech noise environments. We report both intrusive metrics, and perceptual metrics.

Intrusive metrics use both the clean target speech file and the enhanced audio. We use the following intrusive metrics,

- [SISDR](#) as a measure of the amount of noise removed by the algorithm
- [PESQ](#) as a measure of the speech quality of the processed audio
- [STOI](#) as a measure of the speech intelligibility improvement by the algorithm

Perceptual metrics to model the subjective human experience of quality. They are generated by the models described in the [DNSMOS P.835 paper](#) by Microsoft and are reportedly correlated highly with human Mean Opinion Scores. For a detailed explanation about the scale of these metrics, please refer to [this paper](#). Higher scores means higher audio quality. We use the following perceptual metrics,

- OVR measures the overall audio quality
- BAK measures the quality and quantity of residual noise
- SIG measures the speech quality

2.3.1 8ms Latency Model

Please refer to section 2.3 for explanations about the metrics used below.

Intrusive Metrics:

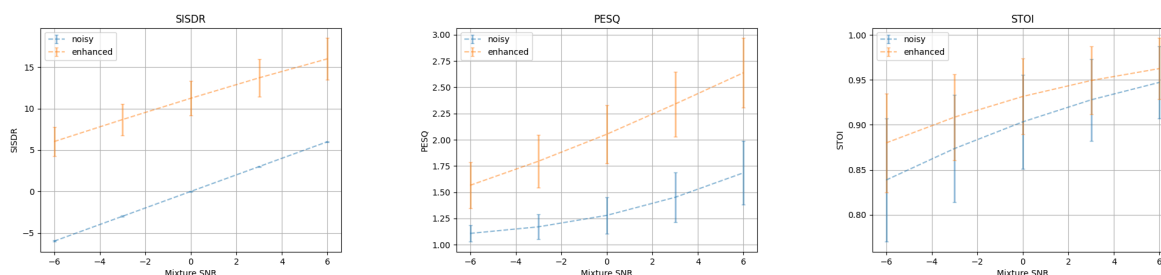


Figure 5: Improvements in intrusive metrics from AINRGP_16khz_4hop_8algo_v4 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

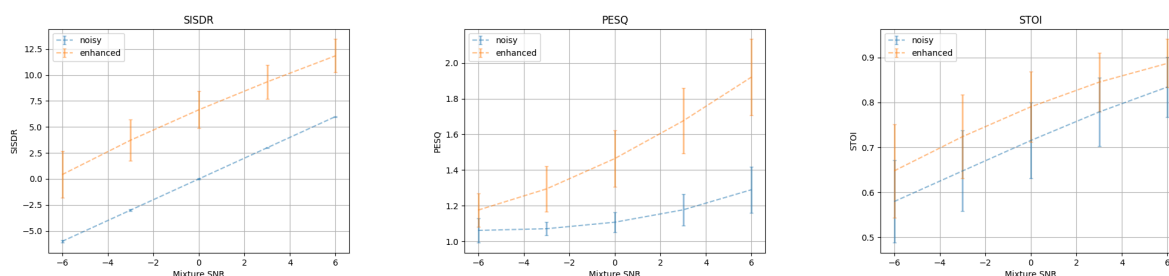


Figure 6: Improvements in intrusive metrics from AINRGP_16khz_4hop_8algo_v4 with a wide variety of speech babble noises from the [WHAM! Dataset](#) across SNR levels

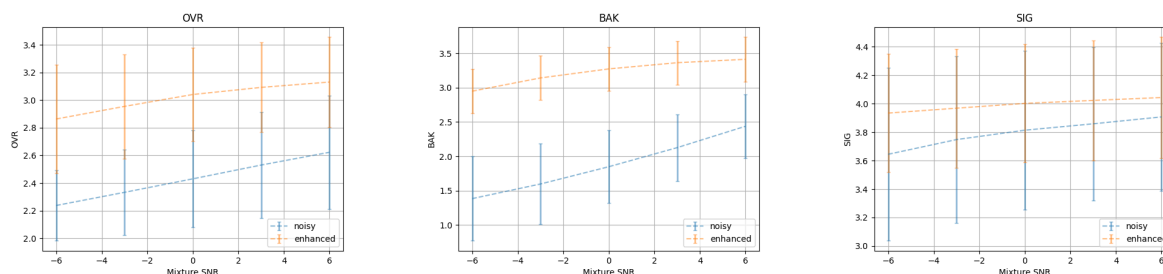
Perceptual Metrics:

Figure 7: Improvements in perceptual metrics from AINRGP_16khz_4hop_8algo_v4 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

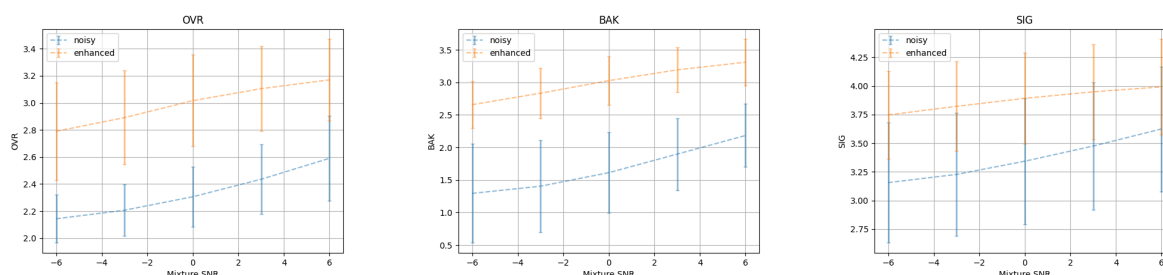


Figure 8: Improvements in perceptual metrics from AINRGP_16khz_4hop_8algo_v4 with a wide variety of speech babble files from the [WHAM! Dataset](#) across SNR levels

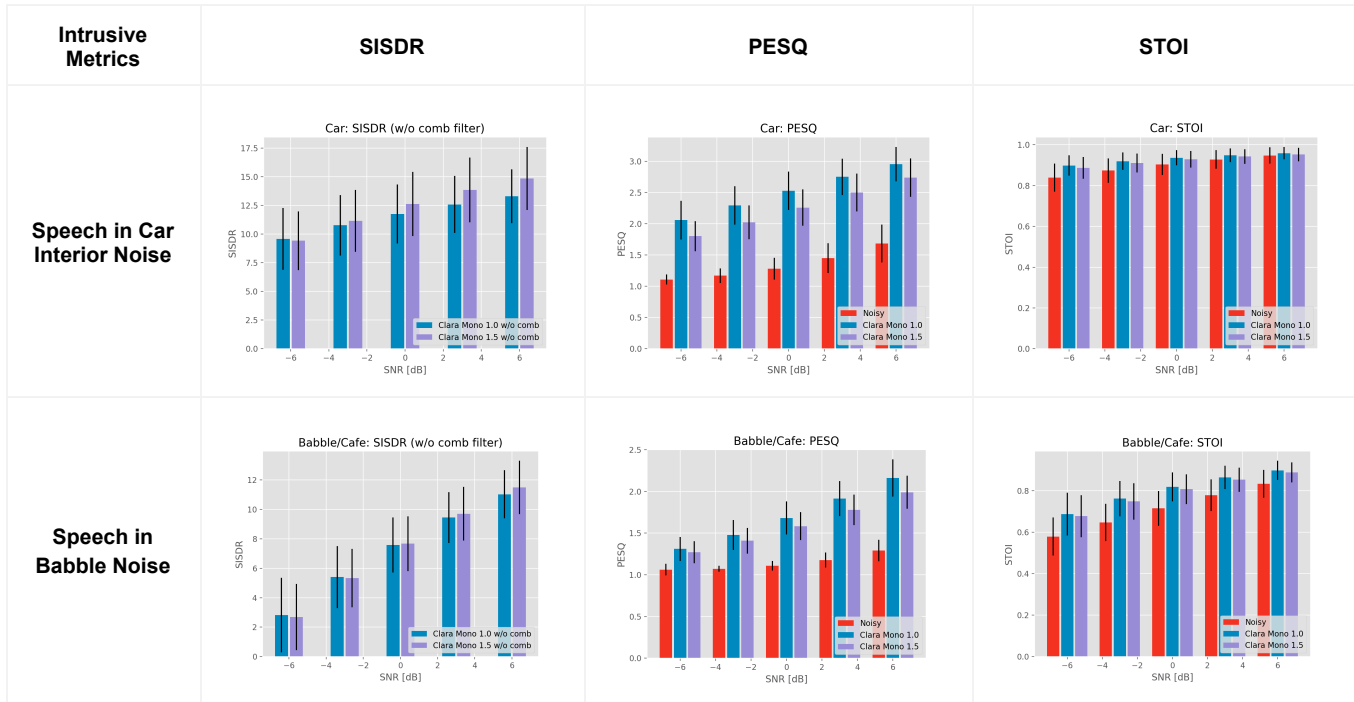
2.3.2 2ms Latency Models (comparison between v2 and v3)

In the figures below, we use the naming convention:

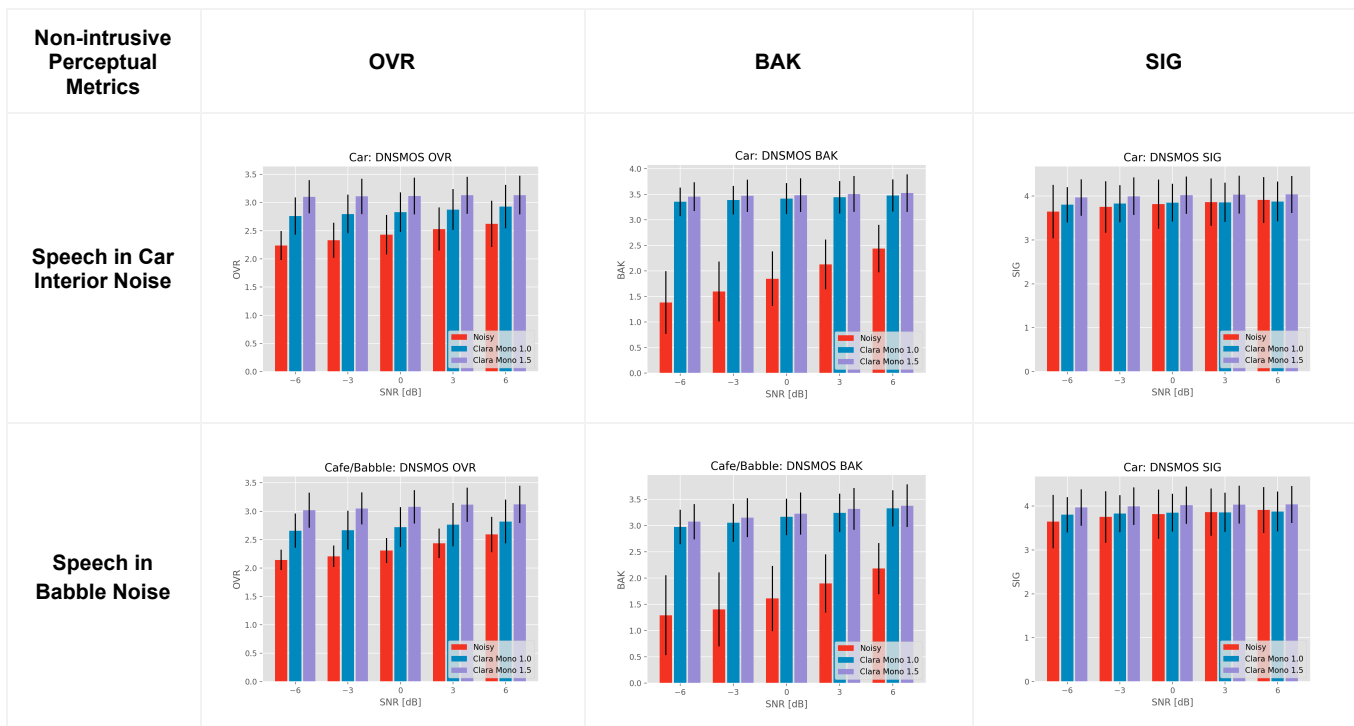
- ClaraMono-2ms v1.0 – AINRGP_16khz_1hop_2algo_v2
- ClaraMono-2ms v1.5 – AINRGP_16khz_1hop_2algo_v3

See section 2.3 for a description of the different metrics. ClaraMono-2ms v1.5 (the latest release of 2ms AINR) has significant improvement for perceptual metrics as measured by DNSMOS. We note that these DNSMOS improvements come at the cost of a slight drop in PESQ and other traditional speech quality metrics. We find that DNSMOS correlates more highly with human evaluation than PESQ.

Intrusive Metrics:

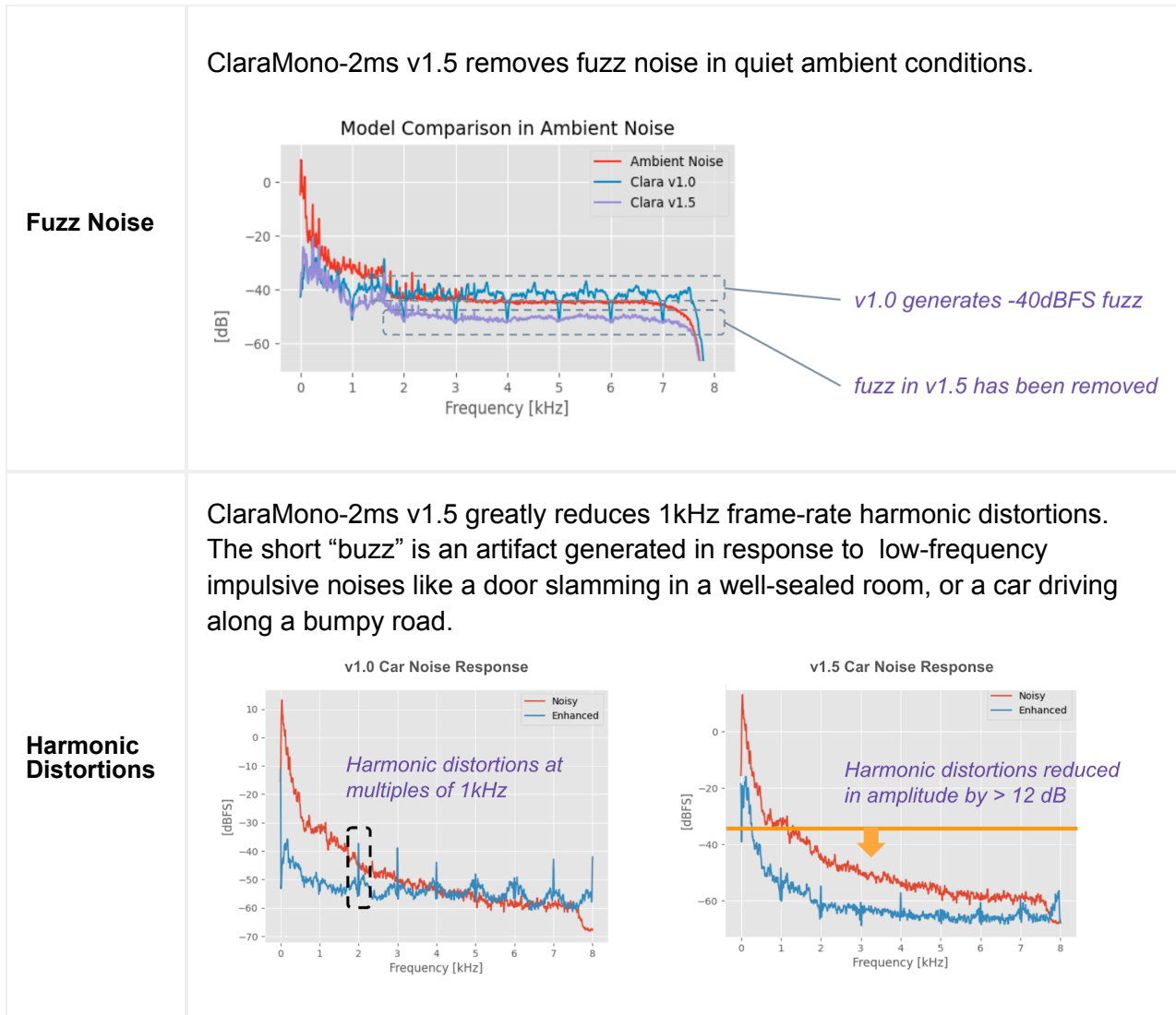


Perceptual Metrics:



Other noise artifacts:

ClaraMono-2ms v1.5 removes fuzz and harmonic distortion found in v1.0



2.4 Spoken Language Understanding (SLU)

The model recognizes a predefined set of intents defined in [Section 1.4](#) and is intended to demonstrate typical “Smart Homes” voice command usages. The model was trained using proprietary data. To demonstrate the flexibility of our approach across languages, we provide two models:

- A model recognizing voice intents in English
- A model recognizing voice intents in Korean

For the evaluation of the model, please refer to the **Quick Start Guide Section 1.2.2.1** for instructions on how to set up the accompanying GUI that visualizes the voice commands in a real-time 3D smart home simulation.

Specifications:

- **Audio In:**
 - 8 kHz Sampling Rate
 - Monaural
 - 16 bits (pcm)
- **Output:** N confidence scores corresponding to N-1 intents and a <Unknown> token

Model Performance:

Quantitative evaluation⁸ of the SLU models will be shared in a future software release.

⁸ We are aware of a relatively high false positive rate in the English SLU, compared to Korean SLU. Future improvements including broad variations of distractors can be implemented to reduce this error.

3. Appendix

3.1 Appendix A

See expanded list of possible Korean commands for smart home demo in the table below. Evaluators can swap in different rooms for light control. Similarly, evaluators can swap in different languages for TV controls.

Control lights	Control lights in the [bathroom/bedroom/kitchen]	Control Volume (for TV & Music)
<ul style="list-style-type: none"> ● 불 켜줘. ● 조명을 켜 주세요. ● 지금 불을 켜. ● 불을 켜주시수 있으세요? ● 불 좀 켜줄래? ● 조명 켜도 되나요? ● 방에 불을 켜 주시겠어요? ● 조명 키는 걸 도와줄 수 있어? ● 불을 키는 게 어때요? ● 조명을 켜주세요 ● 불 꺼줘. ● 조명을 꺼 주세요. ● 지금 불을 꺼. ● 조명을 끌 수 있을까요? ● 불 좀 꺼줄래? ● 조명/불을 꺼도 되나요? ● 방에 불을 꺼 주시겠어요? ● 조명을 끄는 걸 도와줄 수 있어? ● 불을 끄는 게 어때요? ● 조명을 꺼주세요 	<ul style="list-style-type: none"> ● [화장실/침실/부엌] 불 켜줘. ● [화장실/침실/부엌]의 조명을 켜 주세요. ● [화장실/침실/부엌] 불을 켜 ● [화장실/침실/부엌] 조명을 켤 수 있을까요? ● [화장실/침실/부엌] 불 좀 켜줄래? ● [화장실/침실/부엌]에 불 켜도 되나요? ● [화장실/침실/부엌]에 불을 켜 주시겠어요? ● [화장실/침실/부엌] 불을 켜줄 수 있겠니? ● [화장실/침실/부엌] 불을 켜는 게 어때요? ● [화장실/침실/부엌] 조명을 켜주세요 ● [화장실/침실/부엌] 불 꺼줘. ● [화장실/침실/부엌]의 조명을 꺼 주세요. ● [화장실/침실/부엌] 불을 꺼 ● [화장실/침실/부엌] 조명을 꺼줄 수 있어? ● [화장실/침실/부엌] 불 좀 꺼줄래? ● [화장실/침실/부엌] 불을 꺼도 되나요? ● [화장실/침실/부엌]에 불을 꺼 주시겠어요? ● [화장실/침실/부엌] 불 끄는 걸 도와줄 수 있어? ● [화장실/침실/부엌] 불을 끄는 게 어때요? ● [화장실/침실/부엌] 조명을 꺼주세요 	<ul style="list-style-type: none"> ● 소리 올려. ● 소리 높여 줘. ● 소리 좀 올려줄래 ? ● 소리 더 높이고 싶어. ● 더 크게 부탁해 ● 볼륨/음량을 높여줘 ● 소리 레벨 좀 올려 줄래? ● 더 크게 하자. ● 소리 높여 주세요 ● 볼륨을 올려줘 ● 소리 낮춰. ● 소리 낮춰 줘. ● 소리 좀 낮출래? ● 소리 더 낮추고 싶어. ● 더 조용하게 부탁할게 ● 소리 볼륨을 줄여. ● 소리 레벨 좀 낮춰 줄래? ● 낮춰줄래 ● 볼륨을 낮추길 부탁할게 ● 소리를 줄여.
Control TV	Set TV language to [Korean/English/German/Mandarin]	Control Music
<ul style="list-style-type: none"> ● TV 켜줘. ● TV를 켜 주세요. ● TV 좀 켜줄래? ● TV를 켜줄 수 있을까? ● TV 켜도 되나요? ● TV 켜 주시겠어요? ● 누군가 TV를 켜 줄 수 있나요? ● TV 켜는 게 어때요? ● TV 보고 싶어, 켜 줘. ● TV를 제발 켜주세요 ● TV 꺼줘. ● TV를 꺼 주세요. ● TV 좀 꺼줄래? ● 나 대신 TV를 꺼줄 수 있을까? ● TV 꺼도 되나요? ● TV 꺼 주시겠어요? ● 누군가 TV를 꺼 줄 수 있나요? ● TV 끄는 게 어때요? ● TV 본 거 끝났어, 꺼 줘. ● TV를 제발 꺼주세요 	<ul style="list-style-type: none"> ● 언어를 [한국어/중국어/영어/독일어]로 설정해. ● 언어를 [한국어/중국어/영어/독일어]로 설정해 주세요. ● 언어를 [한국어/중국어/영어/독일어]로 설정할 수 있나요? ● 언어를 [한국어/중국어/영어/독일어]로 바꾸고 싶어요. ● 언어 설정을 [한국어/중국어/영어/독일어]로 바꿔 줄래요? ● 언어를 [한국어/중국어/영어/독일어]로 전환해. ● 시스템 언어를 [한국어/중국어/영어/독일어]로 변경해. ● 언어를 [한국어/중국어/영어/독일어]로 설정하는 것이 가능한가요? ● 언어를 [한국어/중국어/영어/독일어]로 설정하고 싶어요. ● 기본 언어를 [한국어/중국어/영어/독일어]로 설정해. 	<ul style="list-style-type: none"> ● 음악 켜. ● 음악 좀 틀어줘. ● 음악 재생해 줄 수 있어? ● 음악 듣고 싶어, 켜 줘. ● 음악 재생 시작해 주세요. ● 내게 음악 좀 틀어 줄래? ● 음악 좀 틀어 봐. ● 음악을 시작해줘 ● 음악 플레이어를 활성화해. ● 음악 듣고 싶어, 켜 줘. ● 음악 꺼. ● 음악 일시정지해. ● 음악 멈춰 줄 수 있어? ● 음악 멈춰 주세요. ● 음악을 멈춰줄래? ● 내 대신 음악 좀 꺼줄래? ● 음악 지금 멈춰. ● 음악 플레이어를 비활성화해. ● 음악 듣기 끝났어, 꺼 줘. ● 음악 재생 중지해.

4. Change Log

Version	Release Date	Description
1.0	2023-04-09	Initial release
1.1	2023-05-03	Add model performance graphs and reports
1.2	2023-07-23	Add metrics for WWDSNIPS_8khz_16ms_v2 (update from v1) Add metrics for AINRGP_16khz_4hop_8algo_v3 (update from v1) Change naming conventions of AINR and WWD models
1.3	2023-10-20	Add GSC_8khz_16ms_v0 with performance metrics Add AINRGP_16khz_1hop_2algo_v0 with performance metrics
1.4	2023-12-28	Add WWDALEXA_8khz_32ms_v0 Add AINRGP_16khz_4hop_8algo_v4 Add AINRGP_16khz_1hop_2algo_v1
1.5	2024-03-18	Add MP chip measurements on EVK2v2 Add AINRGP_16khz_1hop_2algo_v2
1.6	2024-04-29	Add EN-SLU_SH_8khz_16ms_v0 Add KR-SLU_SH_8khz_16ms_v0 Add AINRGP_16khz_1hop_2algo_v3