

Algorithms and Evaluation Guide v1.3

Femtosense includes the following pre-trained algorithms as part of its evaluation kit:

- Single-microphone “Hey Snips” Wake Word Detection
- Single-microphone AI Noise Reduction

Model Name	Description
WWDSNIPS_8khz_16ms_v2	Wake Word Detection “Hey Snips” 1 microphone 8 kHz sampling rate 16 ms hop-size Version 2
GSC_8khz_16ms_v0	Keyword Detection (Google Speech Commands) 1 microphone 8 kHz sampling rate 16 ms hop-size Version 0
AINRGP_16khz_4hop_8algo_v3	AI Noise Reduction “General Purpose” 1 microphone 16 kHz sampling rate 4 ms hop-size 8 ms algorithmic latency Version 3
AINRGP_16khz_1hop_2algo_v0	AI Noise Reduction “General Purpose” 1 microphone 16 kHz sampling rate 1 ms hop-size 2 ms algorithmic latency Version 0

Table of Contents

- 1. Algorithms.....3
 - 1.1 Wake Word Detection Algorithms..... 3
 - 1.2 Multi-Class Keyword Detection (Google Speech Commands)..... 4
 - 1.3 AI Noise Reduction Algorithms..... 5
- 2. Proposed Evaluation..... 6
 - 2.1 Wake Word Detection..... 6
 - 2.2 Multi-Class Keyword Detection (Google Speech Commands)..... 8
 - 2.3.1 8ms Latency Model..... 10
 - 2.3.2 2ms Latency Model..... 12
- 3. Change Log..... 13

1. Algorithms

1.1 Wake Word Detection Algorithms

These wakeword detection algorithms are trained to recognize the phrase “Hey Snips.” 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 model has been trained against indoor environmental noise, competing speech, and room reverberance. The model audio input uses an INT16 PCM format.

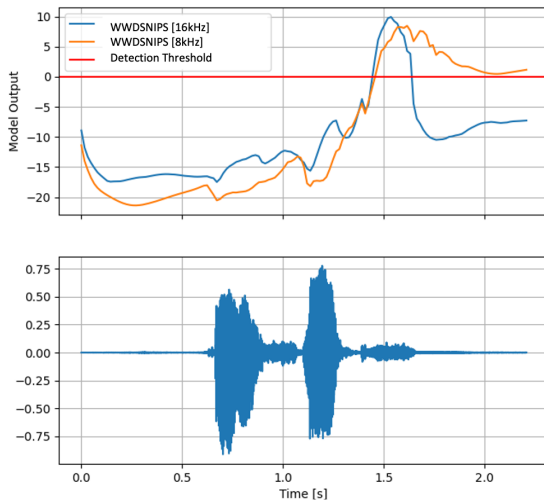
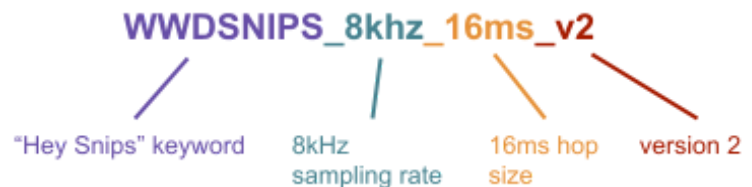


Figure 1: Wakeword detection algorithms return scalar values for each input frame of audio (every 16ms). When the output exceeds the detection threshold (default 0), it is interpreted as a positive detection of the keyword.

Variants exist for 16kHz and 8kHz sampling rates.

Model Naming Convention Example

- Target keyword: “Hey Snips”
- Input audio sampling rate: 8kHz
- Hop size: 16ms
- Algorithm version: 2



Summary of Models

Model Name	Compiler Version	Optimized Firmware	VDD Power Consumption
WWDSNIPS_8khz_16ms_v2	0.2.8	yes	193 μ W

1.2 Multi-Class Keyword Detection (Google Speech Commands)

This model was trained on the [Google Speech Commands Dataset](#), recognizing 11 specified keywords: ["On", "Off", "Left", "Right", "Up", "Down", "Yes", "No", "Stop", "Go", "Hey Snips"]. Notably, "Hey Snips" was incorporated from a separate dataset. Input to the model consists of a sequence of raw waveform frames, which yields an output sequence of probabilities, each indicating the likelihood of a keyword's presence at each frame. To bolster its robustness, other keywords from the Google Speech Commands Dataset serve as "distractors," countering potential false activations. The model has 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.

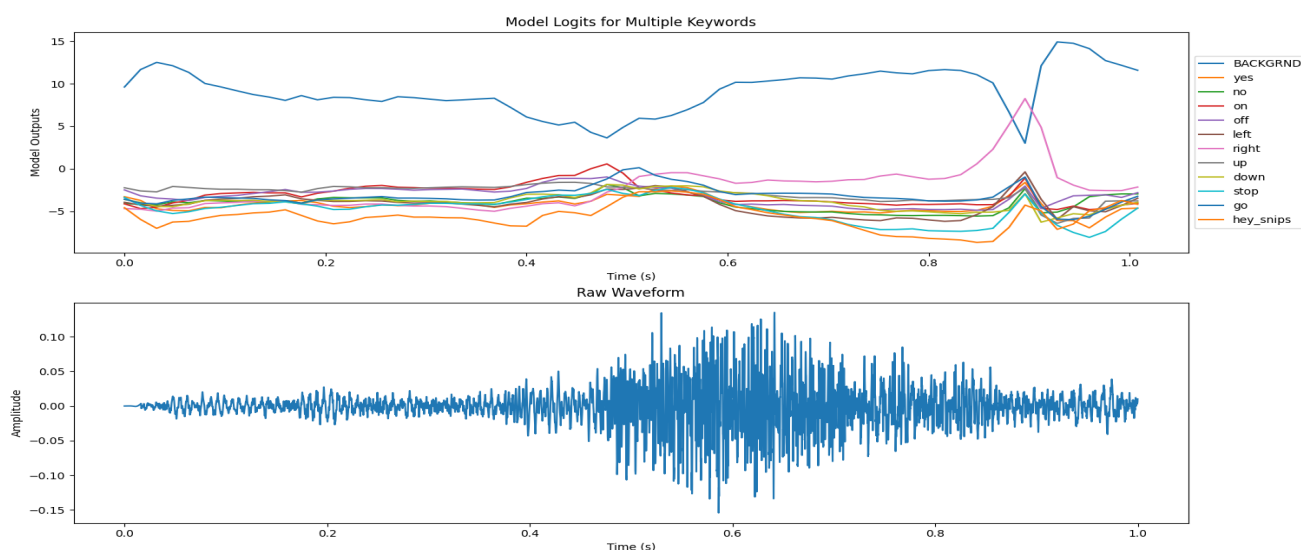
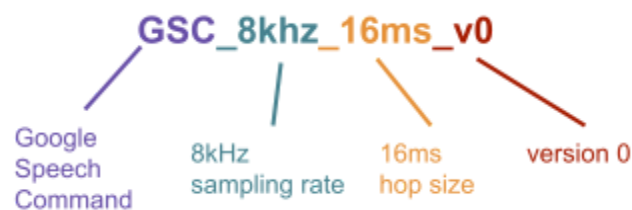


Figure 2: Keyword Detection (class: "right") example using GSC_8KHz_16ms_V0 model. Model outputs represent confidence scores for each keyword class, with peaks indicating likely keyword detection. "BACKGRND" represents the background class.

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



Summary of Models

Model Name	Compiler Version	Optimized Firmware	VDD Power Consumption
GSC_8kHz_16ms_v0	0.2.8	yes	461 μ W

1.3 AI Noise Reduction Algorithms

The general purpose AI Noise Reduction algorithm removes 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. The model input and output audio uses 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: 3

Summary of models

Model Name	Algorithmic Latency	Compiler Version	Optimized Firmware	VDD Power Consumption
AINRGP_16khz_4hop_8algo_v3	8 ms	0.2.8	yes	1.41 mW
AINRGP_16khz_1hop_2algo_v0	2ms	0.2.8	yes	3.42 mW

2. Proposed Evaluation

2.1 Wake Word Detection

Our algorithm detects `Hey Snips` 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: 3dB 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.**

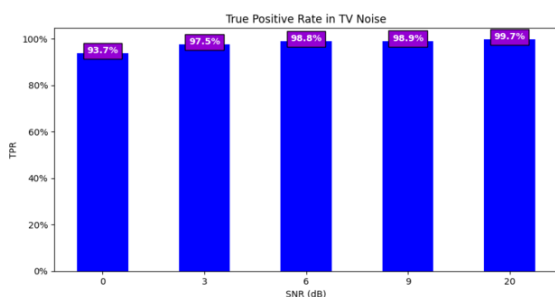


Figure 3: True Positive Rate (TPR) for the WWDSNIPS_8khz_16ms_v2 on a test set of 3,605 samples played with TV background noise from the [TVSM-cuesheet Dataset](#).

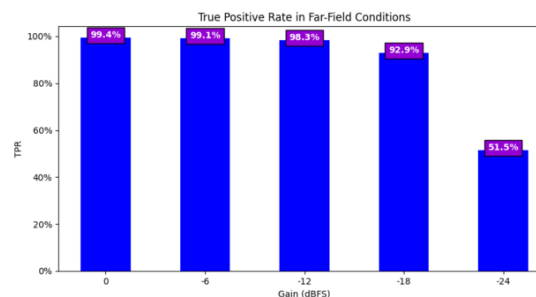


Figure 4: True Positive Rate (TPR) for the WWDSNIPS_8khz_16ms_v2 on a test set of 3,605 samples played in silence at different distances. The reference at 0 dBFS corresponds to 1m distance of the speaker and device, and each added -6dBFS corresponds to doubling the distance between speaker and receiver 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 algorithm removes background noise while preserving the speech. We recommend evaluating the algorithm in the following conditions:

- Noise Environments: car, **babble (restaurant/café background)**, and transient sounds
- Signal to Noise Ratios: The algorithm should provide good performance above -3 dB SNR. The algorithm should work without voice distortions above 0 dB SNR.
- 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

Testing should be conducted in a non-reverberant environment, otherwise the effective SNRs levels will be lower.

2.3.1 8ms Latency Model

Model Performance:

The model works well down to 0dB SNR without voice distortion. At -3dB, the voice is still preserved but the user can expect mild distortion.

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 use 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

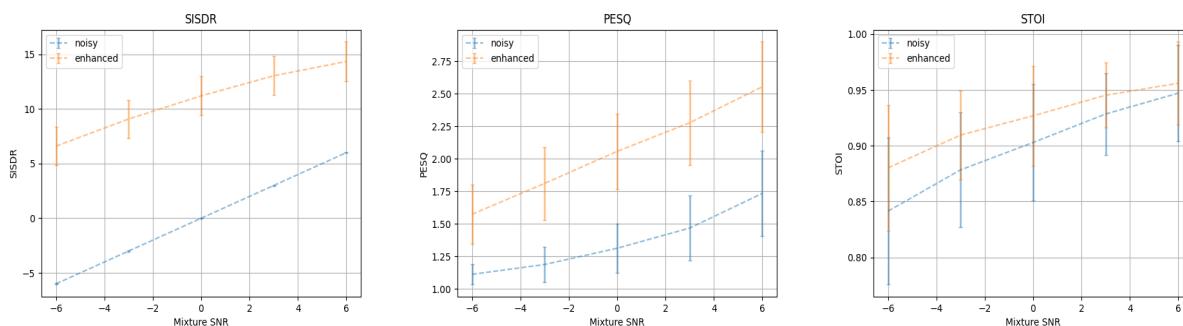


Figure 5: Improvements in intrusive metrics from AINRGP_16khz_4hop_8algo_v3 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

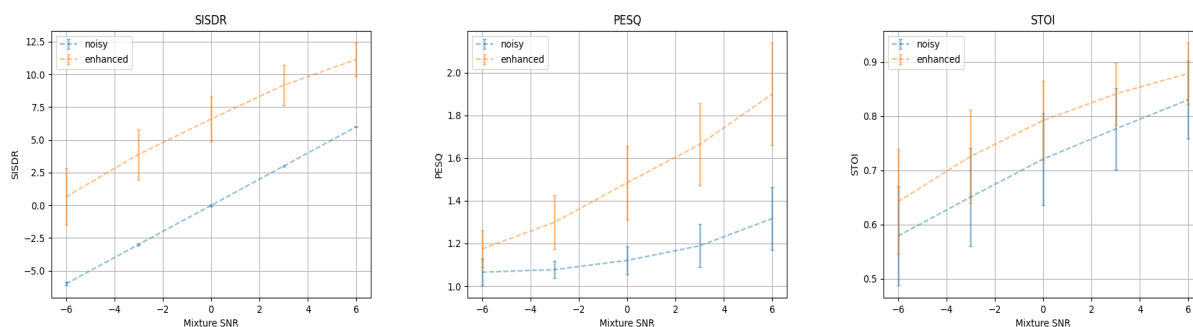


Figure 6: Improvements in intrusive metrics from AINRGP_16khz_4hop_8algo_v3 with a wide variety of speech babble noises from the [WHAM! Dataset](#) across SNR levels

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 amount of noise removal
- SIG measures the speech quality

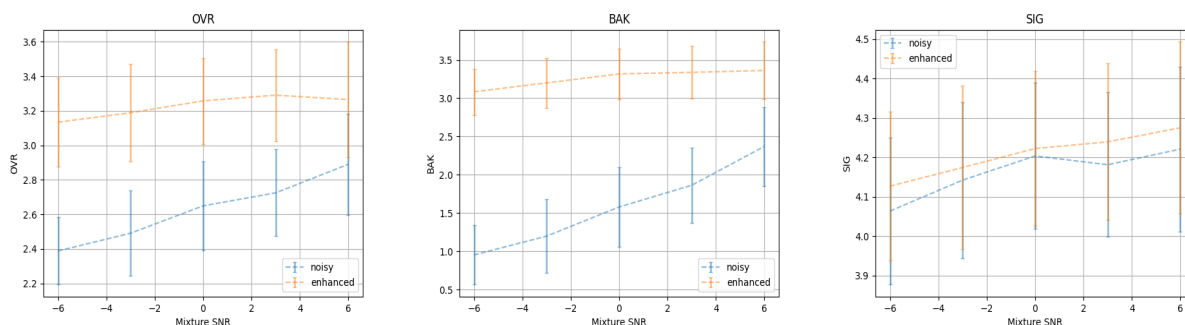


Figure 7: Improvements in perceptual metrics from AINRGP_16khz_4hop_8algo_v3 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

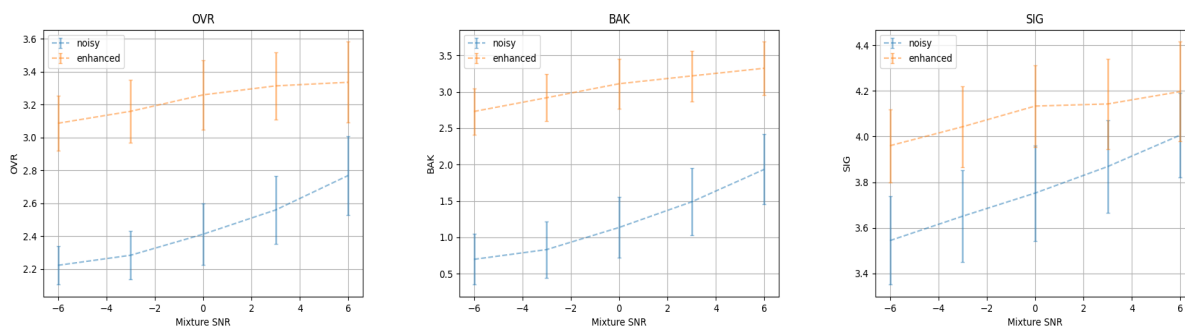


Figure 8: Improvements in perceptual metrics from AINRGP_16khz_4hop_8algo_v3 with a wide variety of speech babble files from the [WHAM! Dataset](#) across SNR levels

2.3.2 2ms Latency Model

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

Intrusive Metrics:

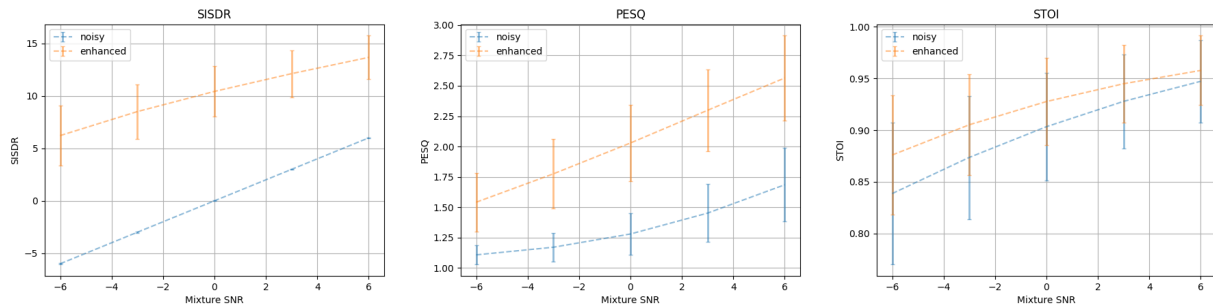


Figure 9: Improvements in intrusive metrics from AINRGP_16khz_1hop_2algo_v0 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

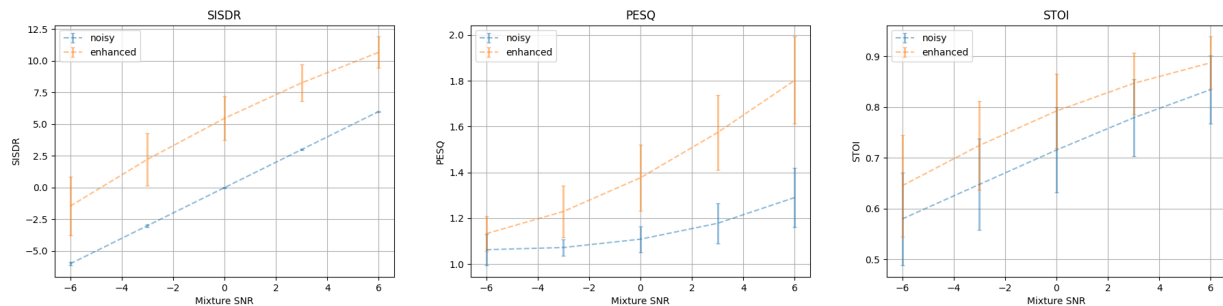


Figure 10: Improvements in intrusive metrics from AINRGP_16khz_1hop_2algo_v0 with a wide variety of speech babble noises from the [WHAM! Dataset](#) across SNR levels

Perceptual Metrics:

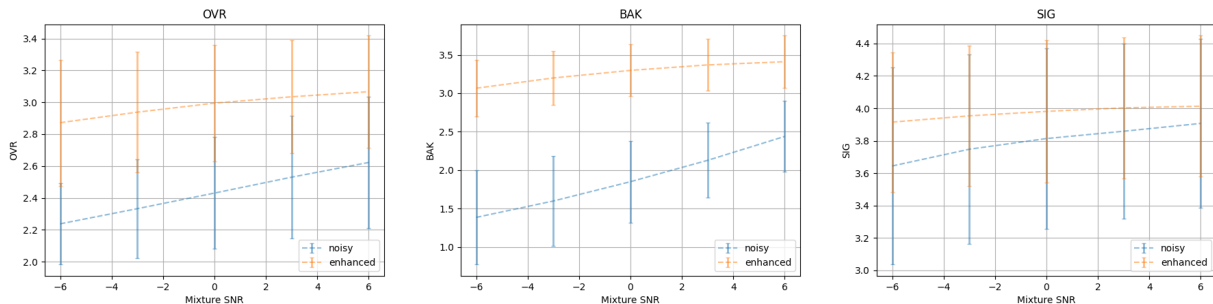


Figure 11: Improvements in perceptual metrics from AINRGP_16khz_1hop_2algo_v0 with a wide variety of car noises from the [Vehicle Interior Sounds Dataset](#) across SNR levels

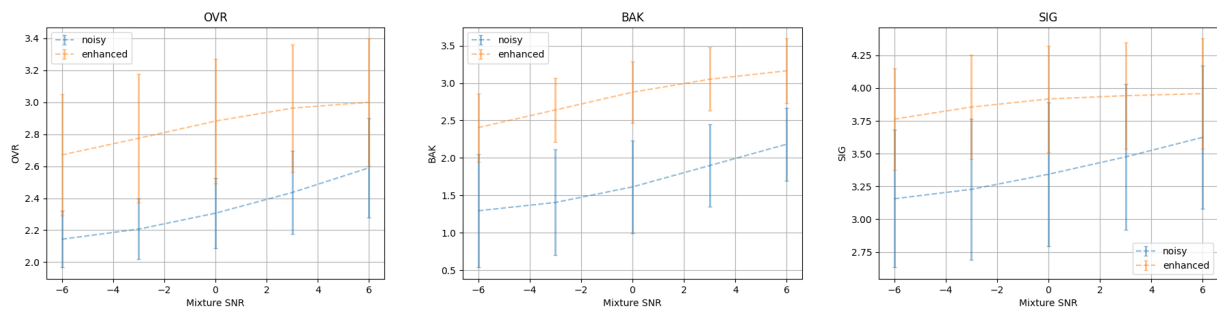


Figure 12: Improvements in perceptual metrics from AINRGP_16kHz_1hop_2algo_v0 with a wide variety of speech babble files from the [WHAM! Dataset](#) across SNR levels

3. 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 WWD SNIPS_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