Multi-Microphone Noise Reduction Techniques for Accurate IoT Acoustic Disdrometer Sensor Systems

Soheyl Faghir Hagh, Jordan Bourdeau, Tian Xia, and Christian Skalka
University of Vermont, Burlington, VT, USA

***Abstract*—This paper presents a lightweight, embedded noise-reduction framework for acoustic precipitation sensing (i.e., rain, hail, snow, rain-on-snow, and sleet) in low-power wide area networks and IoT sensor nodes. A dual-microphone digital platform with MEMS microphones, using spectral subtraction with Wiener refinement implemented on Teensy 4.1, and a single-microphone analog platform using multilevel wavelet denoising implemented on our Aura board, are evaluated at 44.1 kHz as part of the evaluation. Field experiments under windy snow and noisy rain conditions show SNR improvements of up to 16.93 dB for the dual-microphone approach and up to 5.39 dB for wavelet denoising. The results demonstrate that classical, noise-canceling methods enable reliable low-noise data acquisition for wireless acoustic disdrometers.**

***Index Terms*—Acoustic disdrometer, Denoising, IoT, Precipitation acoustics, Spectral subtraction, Wavelet denoising, Wiener filtering**

I. Introduction

Environmental acoustic sensing is increasingly used in remote wireless and Internet of Things (IoT) sensor networks to characterize natural phenomena, monitor ecological activity, and to support machine learning-based classification tasks [1]. In such systems, microphones often operate in uncontrolled outdoor soundscapes where recordings contain the target event mixed with diverse environmental perturbations (wind, rain, biological activity, and human/anthropogenic sounds) [2].

This challenge is especially relevant in impact-based acoustic sensing, such as rainfall/acoustic disdrometry sensor systems, where raindrop impact sounds are processed into features for rainfall occurrence/intensity estimation and related event-driven monitoring [3]. In settings where machine learning supports detection, the accuracy and robustness of learning-based classifiers are strongly influenced by how effectively background interference is suppressed during both model training and deployment [4].

Outdoor acoustic precipitation sensing is highly susceptible to non-precipitation sounds (especially wind and ambient background), which can mask raindrop impacts and produce false detections [5]. Field studies of low-cost acoustic rainfall sensors note that real deployments may record both raindrops and background noise continuously, making robust event detection and filtering essential [6]. Rain is also widely treated as a strong geophonic component in environmental audio analysis, underscoring how precipitation and ambient noise can dominate and distort acoustic measurements [7].

A substantial body of recent literature has examined speech and audio denoising algorithms. Classical approaches such as spectral subtraction, Wiener gain, Minimum Mean Square Error (MMSE) estimation, and wavelet-based denoising methods remain widely used because of their analytical simplicity and well-understood behavior in additive noise environments [8].

Classical statistical filters remain practical baselines for wireless sensor nodes due to low and predictable compute/memory cost, whereas many deep-learning denoisers are resource-intensive and can be difficult to run in real time on constrained edge hardware [9].

Motivated by these considerations, this work focuses on efficient classical denoising techniques such as spectral subtraction, Wiener, and Wavelet filtering that provide practical noise reduction for acoustic precipitation isolation and detection.

To support embedded denoising within our machine learning-based acoustic disdrometer system [10], we develop a multi-microphone acquisition architecture that enables spatial noise analysis and enhances signal quality before feature extraction. Implementing synchronized multichannel recording on resource-limited hardware, such as microcontrollers, introduces challenges related to Analog-to-Digital Converter (ADC) throughput, RAM limitations, SD-card I/O bandwidth, and channel-switching latency, which we address in this work.

In this work, we combine two independent platforms within a unified calibration framework for precipitation sensing: (i) an Arduino-class analog multi-channel audio acquisition library [11] implemented on an Aura board supporting 10-bit data width via rapid channel switching and double-buffered SD logging, and (ii) a digital I^2S MEMS microphone system on the Teensy 4.1 board, providing synchronous 16-bit Pulse Coded Modulation (PCM) audio suitable for acoustic sensing. Specifically, the main contributions of this work are:

- **Low-noise Data Acquisition for Calibration Dataset:** This work presents a calibration setup for collecting low-noise precipitation acoustic data, forming a labeled dataset for machine learning-based detection and classification, with one-time calibration dataset performed against a $Parsivel^2$ disdrometer as ground truth.
- **Acoustic Resonance Chamber:** A 3-D printed pyramid acoustic structure is designed and validated, using thin plastic sheets that resonate under hydrometeor impact. Snow accumulation was observed, mitigable via color or hydrophobic coatings.
- **Embedded Noise-Reduction Pipeline:** Additionally, we evaluate lightweight noise suppression techniques, including widely-used spectral subtraction, Wiener gain, and wavelet-based denoising for deploying on microcontroller

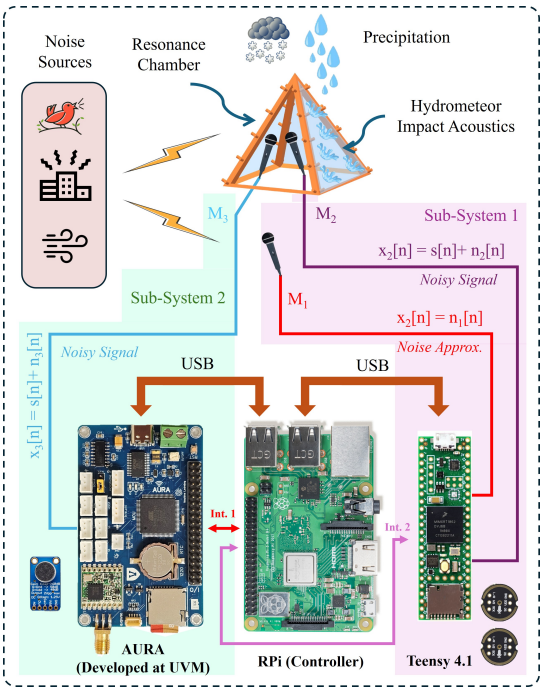

Fig. 1. Data acquisition (DAQ) unit diagram for embedded acoustic disdrometer: Teensy 4.1 with two INMP441 digital MEMS microphones (sub-system 1) and an Aura board with a MAX9814 analog microphone (sub-system 2), both controlled by an RPi 3B+ with interrupts and USB Serial communication.

hardware, demonstrating their feasibility for noise suppression in an acoustic disdrometer application.

- **Signal-to-Noise Ratio Evaluation** The proposed setup is deployed in real-world conditions to capture the acoustics of precipitation (rain and snow) and environmental noise. SNR parameters are calculated before and after noise reduction.

The remainder of this paper is organized as follows: Section II describes the system hardware, Section III identifies noise sources, Section IV investigates the noise reduction methodologies, and Sections V and VI provide the results, discussion, and conclusion.

II. System Hardware

This section describes the hardware in Fig. 1, including the microphone configuration and microcontroller setup. The acoustic disdrometer uses a resonance chamber with dual (M1, M2) and single-microphone (M3) configurations for noise reduction evaluation. An RPi 3B+ is used only for calibration to control the Aura and Teensy boards and upload data. The deployed system uses only the Aura and Teensy boards, with the Aura serving as the controller for the prediction pipeline.

A. Analog Single-Microphone Module

Analog recordings are performed on an ATmega2560-based board (Aura) designed at UVM using a custom Arduino-compatible recording library [11]. Here, we provide a high-level overview of our proprietary library; detailed descriptions of the theory of operation and application programming interfaces are available in the repository documentation.

1) Timing Control: ADC sampling is initiated by a compare-match event generated by the ATmega2560 16-bit hardware timer. The timer is configured to trigger conversions at a frequency equal to the desired per-channel sampling rate multiplied by the number of scanned channels in Eq. 1.

$$f_{\text{timer}} = f_s \, N_{\text{ch}} \tag{1}$$

where f_s is the target sampling rate per channel and N_{ch} is the number of ADC channels that the user chooses to be sequentially sampled ($1 \leq N_{\text{ch}} \leq 16$ for the ATmega2560 and in this study, $N_{\text{ch}} = 1$).

The timer trigger frequency (f_{timer}) and the ADC clock frequency (f_{ADC}) are determined by the per-channel sampling rate provided, but use the CPU clock frequency ($f_{\text{CPU}} = 16$ MHz) when selecting appropriate prescaler and compare values. First, the timer frequency (f_{timer}) is determined from Eq. 1. The remaining timer and ADC parameters are set by satisfying Eq. 2(a)-(c) given the range of possible values for each parameter.

The output compare register (OCR) is chosen as a 16-bit value which minimizes the timer error ($|f_{\text{timer}} - f_{\text{target}}|$) with the selected timer prescaler ($N_{\text{TP}} \in \{1, 8, 64, 256, 1024\}$). The timer prescaler is chosen from these values such that (2a) is satisfied (or less than some predetermined error bound). Then, the ADC prescaler ($N_{\text{ADC}_\text{P}} \in \{2, 4, 8, 16, 32, 64, 128\}$) is selected so (2c) holds.

$$f_{\text{timer}} = \frac{f_{\text{CPU}}}{N_{\text{TP}}\,(1 + \text{OCR})} \tag{2a}$$

$$f_{\text{ADC}} = \frac{f_{\text{CPU}}}{N_{\text{ADC}_\text{P}}} \geq N_{\text{cycles}}\, f_{\text{timer}} \tag{2b}$$

$$N_{\text{ADC}_\text{P}} \leq \frac{N_{\text{cycles}}}{N_{\text{TP}}\,(1 + \text{OCR})} \tag{2c}$$

For ATmega2560, $N_{\text{cycles}} = 13$. The largest N_{ADC_P} satisfying (2c) is selected to maximize cycle time and improve sample precision.

Considering an initial $f_s = 44.1$ kHz ($N_{\text{ch}} = 1$), with $f_{\text{CPU}} = 16$ MHz, $N_{\text{TP}} = 1$ and OCR $= 363$ yields $f_{\text{timer}} = 44,077$ Hz, which closely approximates f_s with 0.05% error. The ADC is configured with $N_{\text{ADC}_\text{P}} = 16$, giving $f_{\text{ADC}} = 1$ MHz and satisfying (2b) for $N_{\text{cycles}} = 13$.

2) Buffering ADC Writes: A double-buffered architecture in the recording library allows ADC samples to be written to one buffer while the other is flushed to the on-board SD card, preventing sample loss when acquisition rate exceeds write throughput. The buffer size is chosen as the minimum value that maintains this condition and depends on the performance of the SD card hardware and software library used.

3) Aura Configuration: Recording was performed using the single-channel ADC at approximately 44.1 kHz sampling frequency, a 4 kB buffer, and a 75 MB/s SD card (32 GB). A

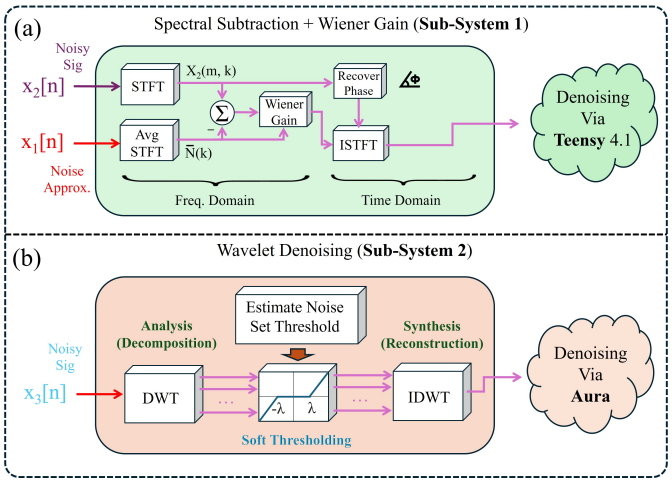

Fig. 2. Processing architecture for precipitation–acoustic noise reduction. The dual-microphone sub-system (a) performs STFT analysis, spectral subtraction, and Wiener refinement using $x_1[n]$ and $x_2[n]$. The single-microphone sub-system (b) applies multilevel wavelet denoising on $x_3[n]$ samples.

maximum sampling rate of 52 kHz was experimentally determined to be achievable with Aura. Standard audio recordings typically use sampling rates of 44.1 kHz or higher to satisfy the Nyquist–Shannon Sampling Theorem.

B. I^2S Dual-Microphone Platform

In addition to ADC-based recording on the Aura platform, a dual-channel I^2S microphone platform is implemented on the Teensy 4.1, which has a 600 MHz clock rate. The I^2S protocol uses SCK, WS, and SD for synchronous PCM audio transmission. Unlike ADC multiplexing, it enables true parallel, time-aligned sampling. Using the Teensy Audio Library, multi-channel audio is recorded simultaneously at 44.1 kHz, 16-bit PCM.

III. Identifying Noise Sources

The system evaluates single- and dual-microphone denoising under static and dynamic noise using two acquisition subsystems. Sub-system 1 (Teensy; Fig. 2(a)) uses a dual-microphone setup: a chamber-mounted microphone for precipitation and noise, and an external microphone only to capture a noise reference. Sub-system 2 (Aura; Fig. 2(b)) uses a single chamber-mounted microphone. This enables direct comparison of single- and dual-mic approaches.

A. Stationary or Quasi-Stationary Noise

Stationary noise arises from microphone self-noise, analog front-end noise, and ADC quantization, forming a broadband noise floor in the absence of precipitation. Teensy recordings show a lower noise floor than Aura and are used for analysis.

B. Nonstationary Noise

Nonstationary noise originates from environmental and structural sources, including wind- and impact-induced vibrations, external acoustics (traffic, machinery, human activity, wildlife, aircraft), wind turbulence, and splash events. These

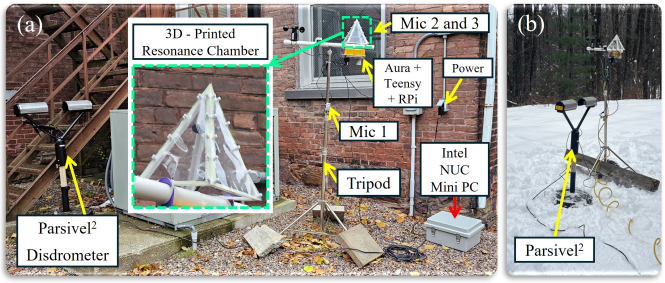

Fig. 3. Deployed calibration setup for precipitation acoustic dataset collection with a Parsivel^2 laser disdrometer providing ground truth measurements for precipitation events (a) Setup Details, (b) Deployment Site (Fayston, VT, USA).

produce time-varying noise, motivating dual-microphone noise suppression techniques.

IV. Noise Reduction Methodologies

Reliable extraction of short impulsive signatures from rain, snow, and hail impacts requires effective suppression of wind and environmental noise. To address this, we investigate two commonly used denoising approaches. The Teensy-based subsystem employs spectral subtraction followed by Wiener filtering to mitigate environmental noise (Fig. 2(a)), while the Aura-based subsystem applies a multilevel discrete wavelet transform to reduce intrinsic microphone noise (Fig. 2(b)). Together, these methods enable a comparative assessment of denoising performance across the two hardware platforms.

A. Sub-system 1 (Dual Microphone Denoising)

In this section, we describe the signal processing pipeline implemented on the Teensy sub-system for a dual-microphone configuration (Microphone 1 and Microphone 2).

1) Short-Time Fourier Transform: Let $x_2[n]$ denote the noisy precipitation signal recorded by Microphone 2 and $x_1[n]$ the noise-reference signal recorded by Microphone 1. The short-time Fourier transform (STFT) of a discrete-time signal $x[n]$ is defined as

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mH]\, w[n]\, e^{-j2\pi kn/N}, \qquad (3)$$

where $w[n]$ is a Hanning window, N is the frame length ($N = 32$ samples), H is the hop size ($H = 4$ samples), m is the time-frame index, and k is the frequency-bin index. Applying Eq. (3) to $x_2[n]$ and $x_1[n]$ produces the STFTs $X_2(m,k)$ and $X_1(m,k)$, respectively.

Because the recorded microphone signals are real-valued, the STFT is implemented using a one-sided frequency representation, such that $k \in \{0, 1, \ldots, K-1\}$ with $K = N/2+1$. These spectra form the basis of the dual-channel denoising method described in the following subsections.

2) Dual-Microphone Spectral Subtraction: The reference signal $x_1[n]$ captures ambient environmental noise without precipitation impacts (see Fig. 3). The magnitude of its STFT

is averaged across all time frames to form a per-frequency noise estimate in Eq. 4.

$$\bar{N}(k) = \frac{1}{M} \sum_{m=0}^{M-1} |X_1(m,k)|, \tag{4}$$

where M is the total number of STFT frames. This operation yields a stationary noise magnitude estimate for each frequency bin.

Spectral subtraction is then applied to the noisy precipitation channel by subtracting the estimated noise magnitude from the magnitude spectrum of $X_2(m,k)$,

$$\hat{S}(m,k) = \max\Big(|X_2(m,k)| - \bar{N}(k),\, 0\Big), \tag{5}$$

where the $\max(\cdot, 0)$ operator prevents negative magnitudes and suppresses frequency bins dominated by noise.

3) Wiener Gain Refinement: To further enhance precipitation acoustics, Wiener refinement is applied to the spectrally subtracted magnitude in Eq. (5). The estimated signal and noise power spectra are defined as

$$P_{\hat{S}}(m,k) = \hat{S}(m,k)^2, \qquad P_N(k) = \bar{N}(k)^2, \tag{6}$$

where $P_{\hat{S}}(m,k)$ denotes the estimated precipitation signal power obtained after spectral subtraction and $P_N(k)$ denotes the noise power estimated from the reference microphone.

Using these estimates, the Lim–Oppenheim parametric Wiener gain is computed as in Eq. 7:

$$G(m,k) = \left(\frac{P_{\hat{S}}(m,k)}{P_{\hat{S}}(m,k) + \beta P_N(k) + \varepsilon} \right)^{\alpha}, \tag{7}$$

where β controls the noise suppression strength, α adjusts the attenuation characteristic of the gain, and ε is a small positive constant introduced for numerical stability. In this study, $\alpha = 12$ and $\beta = 9$.

The Wiener-refined magnitude spectrum is obtained as

$$\hat{S}_{\mathrm{W}}(m,k) = G(m,k)\, \hat{S}(m,k). \tag{8}$$

4) Inverse STFT and Time-Domain Reconstruction: The enhanced complex STFT is reconstructed by combining the refined magnitude $\hat{S}_{\mathrm{W}}(m,k)$ with the original phase of the noisy precipitation signal,

$$\hat{X}_{\mathrm{SSW}}(m,k) = \hat{S}_{\mathrm{W}}(m,k)\, e^{j\angle X_2(m,k)}. \tag{9}$$

The time-domain signal $\hat{s}_{\mathrm{SSW}}[n]$ is obtained via inverse STFT using overlap–add (OLA) synthesis with the same window $w[n]$ and hop size H. Let $\hat{x}_m[n]$ denote the length-N inverse discrete Fourier transform of the m-th enhanced STFT frame. The OLA reconstruction is given in Eq. 10 as:

$$\hat{s}_{\mathrm{SSW}}[n] = \frac{\sum_{m=0}^{M-1} \hat{x}_m[n]\; w[n - mH]}{\sum_{m=0}^{M-1} w^2[n - mH]}, \tag{10}$$

where the denominator provides the standard normalization required for exact reconstruction under the nonzero overlap–add (NOLA) condition [13].

B. Sub-system 2 (Single Microphone Denoising)

In this section, we describe the signal processing on the single-microphone Aura sub-system.

1) Discrete Wavelet Transform (DWT): Wavelet denoising is used for the single-channel signal $x_3[n]$. Because precipitation impacts produce sharp, localized increases in amplitude, their energy is concentrated in detail coefficients across scales.

The DWT decomposition uses an orthonormal Coiflet wavelet. In a preliminary evaluation, multiple wavelet families were explored, including Daubechies, Symlets, and Coiflets; the Coiflet family provided the best performance for transient preservation and noise suppression in this implementation. The approximation and detail coefficients at level ℓ are computed according to Eqs. 11 and 12:

$$A_\ell[k] = \sum_n A_{\ell-1}[n]\, h[2k - n], \tag{11}$$

$$D_\ell[k] = \sum_n A_{\ell-1}[n]\, g[2k - n], \tag{12}$$

with $A_0[n] = x_3[n]$, where $h[n]$ and $g[n]$ are the low- and high-pass analysis filters, n is the sample index, k the downsampled coefficient index, and ℓ the decomposition level (up to 15 in this work). $A_\ell[k]$ and $D_\ell[k]$ denote the low-frequency approximation and high-frequency detail wavelet coefficients, respectively.

2) Universal Thresholding: Noise variance is estimated using the finest-scale (highest-frequency) detail coefficients $D_1[k]$ as shown in Eq. 13:

$$\hat{\sigma} = \frac{\mathrm{median}(|D_1[k]|)}{0.6745}. \tag{13}$$

In the implementation, threshold selection follows the minimax rule. For a detail subband of length N, the minimax threshold is defined in Eq. 14:

$$\lambda = \hat{\sigma}\,(0.3936 + 0.1829 \log_2 N)\,, \tag{14}$$

for $N > 32$ (with a universal-threshold fallback for short lengths). The threshold can further be scaled by a user-defined strength factor.

Soft thresholding is applied to each detail coefficient sequence according to Eq. 15:

$$\tilde{D}_\ell[k] = \mathrm{sign}(D_\ell[k]) \max\big(|D_\ell[k]| - \lambda,\, 0\big), \tag{15}$$

3) Wavelet Reconstruction (IDWT): Signal reconstruction uses the synthesis filters $\tilde{h}[n]$ and $\tilde{g}[n]$ as defined in Eq. 16:

$$A_{\ell-1}[n] = \sum_k A_\ell[k]\tilde{h}[n - 2k] + \sum_k \tilde{D}_\ell[k]\tilde{g}[n - 2k]. \tag{16}$$

The final denoised single-microphone output is obtained after full reconstruction to level $\ell = 0$ and is denoted by $s_{\mathrm{wavelet}}[n] = A_0[n]$.

This multiscale formulation efficiently suppresses broadband noise while preserving the sharp transients characteristic of precipitation impacts.

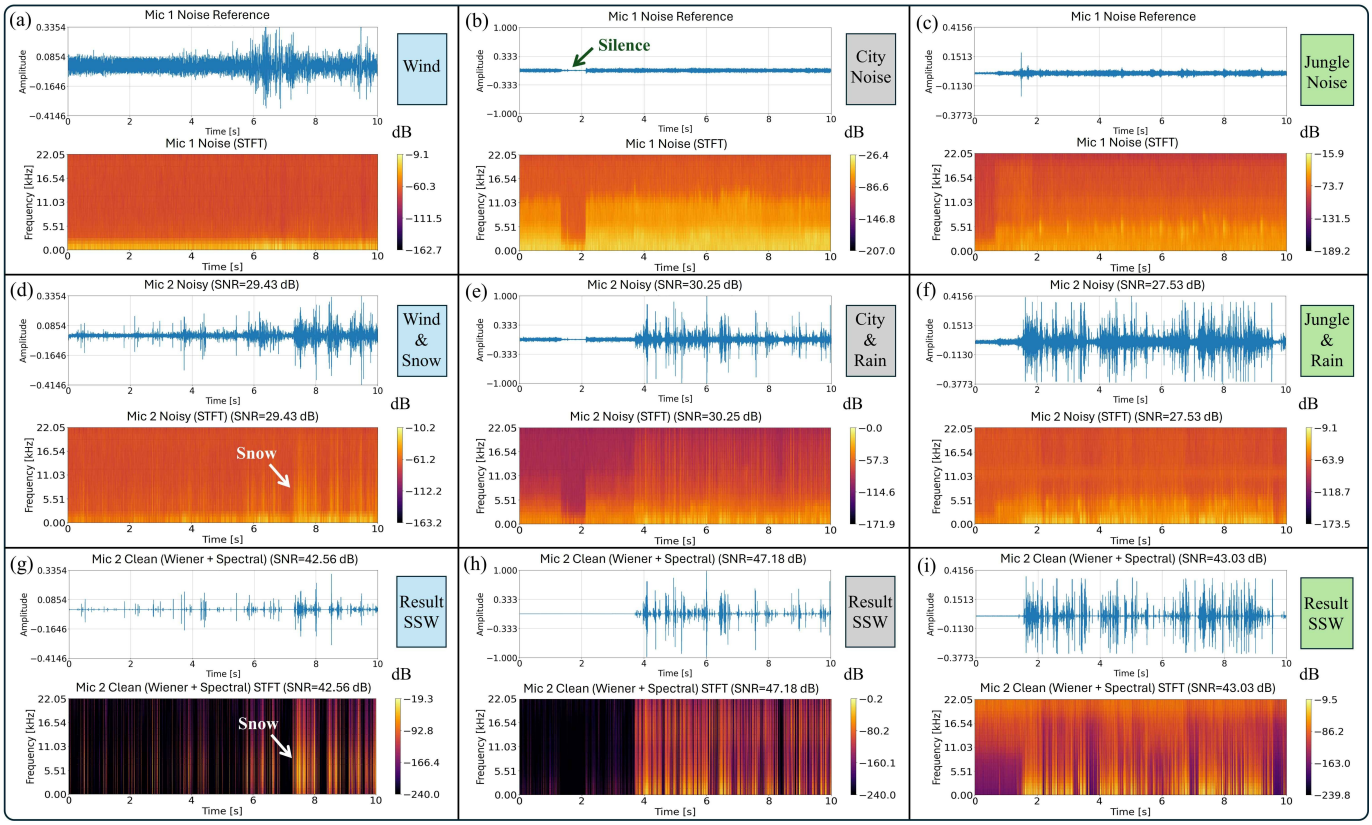

Fig. 4. Spectral subtraction with Wiener gain results (Teensy-based multi-microphone) with the time domain and STFT spectrograms under three conditions: (i) windy snow, (ii) simulated rain with city noise, and (iii) simulated rain with jungle and bird noise. Panels show: (a), (b), (c) noise reference from Mic 1; (d), (e), (f) noisy signals from Mic 2; and (g), (h), (i) corresponding denoised outputs. Color bars represent magnitudes of $X(m,k)$ or STFTs.

V. Results and Discussion

In this section, we describe the data and denoising results obtained from implementing the two techniques.

A. SNR Evaluation

Signal-to-noise ratio (SNR) is computed on a per-event basis, where the signal corresponds to impulsive precipitation-induced impacts that span a broad frequency band. The precipitation event (signal) power P_{Event} is estimated over detected impulse regions, while the noise power P_{Noise} is obtained from the noise approximation channel in the dual-microphone setup (via RMS estimation), or from signal-absent regions in the single-channel case. The SNR is defined in Eq. 17.

$$\text{SNR} = 10\log_{10}\left(\frac{P_{\text{Event}}}{P_{\text{Noise}}}\right). \tag{17}$$

B. Noise Simulation and Acoustics

Three scenarios are evaluated: urban noise, jungle noise, and wind. Urban and jungle noises are played through a nearby speaker, rain is simulated using a porous container with poured water, and wind and snow are collected from real environmental recordings.

C. Spectral Subtraction and Wiener Gain Denoising

The spectral subtraction and Wiener gain (SSW) approach is most effective for suppressing *spatially correlated* noise captured similarly by both microphones [12], including urban background noise, vehicles, wind, and natural environmental sounds (Fig. 4(a)–(i)). By leveraging shared noise statistics between the signal and reference channels, SSW achieves substantial SNR improvements of 13.13 dB for windy snow, 16.93 dB for rain with urban noise, and 15.50 dB for rain with jungle noise (Table I), demonstrating strong performance under diverse and highly nonstationary acoustic

D. Wavelet Denoising

Multi-level wavelet denoising supports single-microphone operation under power or bandwidth constraints (Fig. 5(a)–(f)). While less effective than the dual-microphone SSW approach, it provides consistent SNR improvements of 4.39 dB for windy snow, 5.39 dB for rain with urban noise, and 5.30 dB for rain with jungle noise (Table I), offering a practical trade-off between system simplicity and denoising performance.

E. Limitations

The single-microphone approach exhibits higher noise and reduced denoising efficiency. In the dual-channel configuration, the noise-reference microphone may capture attenuated

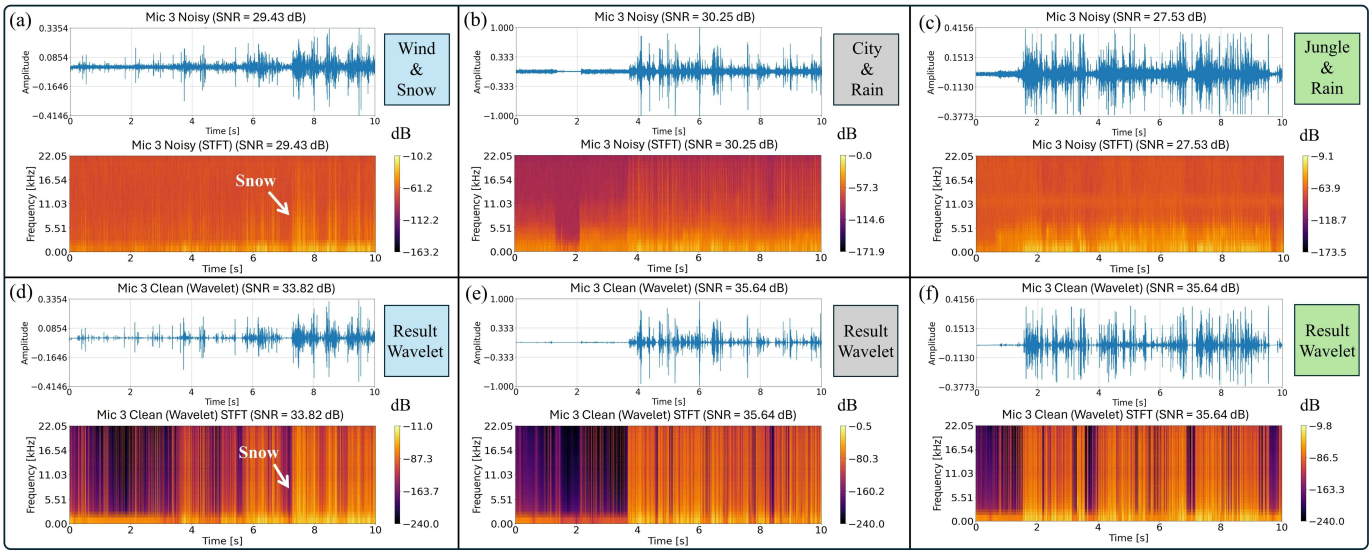

Fig. 5. Multi-level wavelet denoising results (single-microphone via Aura) with the time domain and STFT spectrograms under three conditions: (i) windy snow, (ii) simulated rain with urban acoustic noise, and (iii) simulated rain with jungle acoustic noise. Noise sources, signal levels, and audio samples are identical to Fig. 4. Panels show: (a–c) noisy recordings for windy snow, urban rain, and jungle rain, respectively, and (d–f) corresponding wavelet-filtered outputs.

TABLE I
MEDIAN SNR ENHANCEMENT (DB) ACROSS NOISE ENVIRONMENTS

Method	Before (dB)	After (dB)	ΔSNR (dB)
SSW[a] (Windy Snow)	29.43	42.56	+13.13
SSW[a] (Rain–City)	30.25	47.18	+16.93
SSW[a] (Rain–Jungle)	27.53	43.03	+15.50
WD[b] (Windy Snow)	29.43	33.82	+4.39
WD[b] (Rain–City)	30.25	35.64	+5.39
WD[b] (Rain–Jungle)	27.53	32.83	+5.30

[a] SSW: spectral subtraction with Wiener gain (Teensy sub-system).
[b] WD: wavelet denoising (Aura sub-system).

precipitation events, leading to partial subtraction; however, these components are significantly weaker than the primary signal and do not substantially distort the measurement.

VI. CONCLUSION

This work presents a low-noise framework for acquiring acoustic precipitation data using single- and dual-microphone noise reduction on embedded platforms. Spectral subtraction with Wiener gain and wavelet denoising improves SNR under real-world conditions. The resulting dataset and pipeline support deployment of a fully embedded system calibrated against a commercial disdrometer. Despite the mentioned limitations, the method provides a reliable and practical solution.

ACKNOWLEDGMENT

The materials in this work are supported by the Cooperative Institute for Research to Operations in Hydrology (CIROH) award number AWD00001596 (Corresponding author: Christian Skalka).

REFERENCES

[1] R. Avanzato and F. Beritelli, "An innovative acoustic rain gauge based on convolutional neural networks," *Inf.*, vol. 11, no. 4, 2020.
[2] Y. Hou, et al., "Acoustic sensor-based soundscape analysis and acoustic assessment of bird species richness in Shennongjia National Park, China," *Sensors*, vol. 22, no. 11, 2022.
[3] S. Hwang, et al., "Rainfall observation leveraging raindrop sounds acquired using waterproof enclosure: Exploring optimal length of sounds for frequency analysis," *Sensors*, vol. 24, no. 13, 2024.
[4] W. Mu, et al., "Environmental sound classification using temporal-frequency attention based convolutional neural network," *Sci. Rep.*, vol. 11, 2021.
[5] R. Paškauskas et al., "High-resolution raindrop counting via instantaneous frequency sensing on hydrophobic elastic membranes," *PLOS ONE*, vol. 19, no. 9, p. e0311995, 2024.
[6] N. Peleg et al., "The potential use of low-cost acoustic sensors to detect rainfall for short-term urban flood warnings," *Nat. Hazards Earth Syst. Sci.*, vol. 23, no. 3, pp. 1233–1248, 2023.
[7] C. Sánchez-Giraldo et al., "Ecoacoustics in the rain: Understanding acoustic indices under the most common geophonic source in tropical rainforests," *Remote Sens. Ecol. Conserv.*, vol. 6, no. 4, pp. 474–488, 2020.
[8] S. T. Yousif and B. M. Mahmmod, "Speech enhancement algorithms: A systematic literature review," *Algorithms*, vol. 18, no. 5, 2025.
[9] P. Gairí et al., "Environmental sound recognition on embedded devices using deep learning: a review," *Artif. Intell. Rev.*, vol. 58, Art. no. 163, 2025.
[10] S. F. Hagh et al., "Embedded IoT System for Acoustic Precipitation Phase Partitioning via Edge ML and MFCCs," IEEE Internet Things J., vol. 12, no. 24, pp. 52675–52686, 2025.
[11] J. Bourdeau, "Chrispy: High-performance multichannel ADC sampling and audio recording library," GitHub repository, 2025. [Online]. Available: https://github.com/jbourds/chrispy
[12] R. Kumar, et al., "Single - channel speech enhancement using implicit Wiener filter for high - quality speech communication," Int. J. Speech Technol., vol. 25, no. 3, pp. 745–758, 2022.
[13] SciPy Developers, "scipy.signal.istft - Inverse Short-Time Fourier Transform," SciPy Documentation. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.istft.html. Accessed: Jan. 2, 2026.