LOCUS – LOcalization with Channel Uncertainty and Sporadic Energy

Subrata Biswas
Worcester Polytechnic Institute
Worcester, USA
sbiswas@wpi.edu

Mohammad Nur Hossain Khan
Worcester Polytechnic Institute
Worcester, USA
mkhan@wpi.edu

Alex Colwell
Worcester Polytechnic Institute
Worcester, USA
ascolwell@wpi.edu

Jack Adiletta
Worcester Polytechnic Institute
Worcester, USA
jtadiketta@wpi.edu

Bashima Islam
Worcester Polytechnic Institute
Worcester, USA
bislam@wpi.edu

Abstract

Accurate sound source localization (SSL), such as direction-of-arrival (DoA) estimation, relies on consistent multichannel data. However, batteryless systems often suffer from missing data due to the stochastic nature of energy harvesting, degrading localization performance. We propose *LOCUS*, a deep learning framework that recovers corrupted features in such settings. *LOCUS* integrates three modules: (1) Information-Weighted Focus (InFo) to identify corrupted regions, (2) Latent Feature Synthesizer (LaFS) to reconstruct missing features, and (3) Guided Replacement (GRep) to restore data without altering valid inputs. *LOCUS* significantly improves DoA accuracy under missing-channel conditions, achieving up to 36.91% error reduction on DCASE and LargeSet, and 25.87−59.46% gains in real-world deployments. We release a 50-hour multichannel dataset to support future research on localization under energy constraints. Our code and data available at: https://bashlab.github.io/locus_project/

CCS Concepts

• **Computer systems organization** → **Sensors and actuators**.

Keywords

Sound Source Localization (SSL), Feature recovery, Batteryless embedded systems, Deep Neural Network (DNN).

1 Introduction

Direction-of-arrival (DoA) estimation enables sound source localization (SSL) by extracting spatial and temporal cues from multichannel microphone data. Classical signal processing methods [11, 41] and deep learning approaches [8, 47] have been applied to DoA estimation, with deep learning offering better robustness to real-world challenges like multipath and reverberation.

Intermittently powered batteryless systems face challenges in acquiring full-channel data due to the stochastic nature of energy harvesting. These systems operate using capacitors that store harvested energy, enabling brief active phases followed by sleep when energy depletes. While capacitor size can control these cycles, longer events often span multiple sleep phases, causing data loss [32]. Since DoA estimation requires consistent multichannel input, these interruptions severely degrade localization accuracy. As a result, batteryless systems are typically limited to short classification tasks [23, 29, 32],

and rarely applied to DoA estimation. Still, their low-power operation holds strong potential for applications like environmental monitoring and wildlife tracking, where source localization is essential.

Existing methods for recovering missing data in audio streams rely on statistical imputation—mean [2], hot deck [26], or multiple imputations [38]—but suffer from oversimplification, mismatch sensitivity, or high computational cost [22]. Mutual information-based techniques [46] can detect corrupted regions but fail to recover features effectively for DNN-based DoA estimation.

Recent DNN-based imputation pipelines for batteryless systems [32] address data recovery but are limited to single-channel input, leaving multichannel recovery for DoA estimation an open challenge. Instead of reconstructing raw audio, prior works suggest that recovering key feature representations—such as generalized cross-correlation (GCC) and Mel– Frequency Cepstral Coefficients (MFCC) is more power-efficient and preserves spatial localization cues [10, 18]. This approach is particularly well-suited for batteryless devices, where transmission cost dominates power consumption.

Batteryless devices offer a compelling solution for long-term, maintenance-free sensing in challenging or resource-constrained settings. Motivated by the need for spatial awareness in such scenarios, we design our system to perform DoA estimation under the energy and data constraints typical of batteryless platforms, making spatial intelligence feasible where it was previously impractical. To enable robust DoA estimation in batteryless deployments with intermittent multichannel data loss, we propose *LOCUS*, an offline deep learning framework that reconstructs corrupted multichannel features for sound source localization. Rather than recovering raw audio—which is power-intensive to transmit and process—*LOCUS* focuses on reconstructing key feature representations.

LOCUS comprises three modules: (1) Information-Weighted Focus (InFo), which estimates entropy to locate unreliable features; (2) Latent Feature Synthesizer (LaFS), which reconstructs missing values based on learned relationships; and (3) Guided Replacement (GRep), which replaces only corrupted elements while preserving valid data. The reconstructed features are then passed to a downstream DNN for localization. While the components used in *LOCUS* are studied separately in previous literature, their integration under the same modular framework enables existing SSL models to function robustly under non-ideal conditions, without requiring

changes to SSL model architecture. Our contributions could be summarized as follows.

- We propose *LOCUS*, a three-stage framework that recovers corrupted multichannel features using entropy-guided interpolation, enabling robust localization in batteryless systems.
- We evaluate *LOCUS* across audio and non-audio tasks—on benchmark datasets, and real-world solar and RF-powered environments—and release *LargeSet*, a 50-hour multichannel dataset for future research.

While many approaches address sparse data recovery, *LOCUS* is unique in targeting feature-domain recovery under real-world spatial and temporal dropout in multichannel arrays. Unlike compressed sensing or matrix completion methods, *LOCUS* supports partial observability, asynchronous channels, and energy-constrained operation. Its physically inspired attention modules capture spatial cues, which are essential for sound localization capabilities and are not modeled by general-purpose recovery methods.

LOCUS achieves up to 21.27% and 17.14% lower DoA error than *SoundSeive* [32] on DCASE [36] and LargeSet, respectively, and improves real-world performance by 25.87–59.46% under stochastic dropout. It also reduces 90th percentile localization error by 25.52% in Wi-Fi CSI-based distance estimation [3]. *LOCUS* is well-suited for low-power deployments such as smart buildings and wildlife monitoring, where spatial feature recovery—rather than waveform reconstruction—is sufficient for accurate localization.

2 Missing Data in Batteryless Microphone Arrays

Missing Channels in Intermittent Systems. Batteryless systems powered by RF, solar, or vibration harvesters often experience unpredictable power fluctuations, leading to data loss in multichannel microphone arrays. Environmental dynamics—like human movement, light occlusion, or mechanical interference—cause inconsistent power, resulting in intermittent microphone failures. In distributed setups, spatial variations in harvested energy lead to asynchronous channel availability, while monolithic systems may deactivate subsets of microphones to conserve power. Although full dropout is possible, partial and asynchronous channel loss is far more common and presents a recurring challenge for localization. As shown in Section 7, these power variations significantly degrade localization accuracy. *LOCUS* addresses this by reconstructing features and estimating source locations offline, making it ideal for batteryless and low-power systems without requiring on-device inference.

Quantifying Missing Data (MDP). We define a channel as the full signal from a single microphone in a microphone array. A channel is considered missing if its entire data stream is absent due to energy loss or simulated dropout. To quantify missingness, we define the Missing Data Percentage (MDP) as a temporal metric, where the fraction of time during which one or more channels are unavailable:

$$MDP = \frac{t_1 + t_2 + \cdots + t_n}{T} \times 100$$

where $t_1, t_2, \ldots, t_n$ are the time segments with one or more missing channels, and T is the total audio duration.
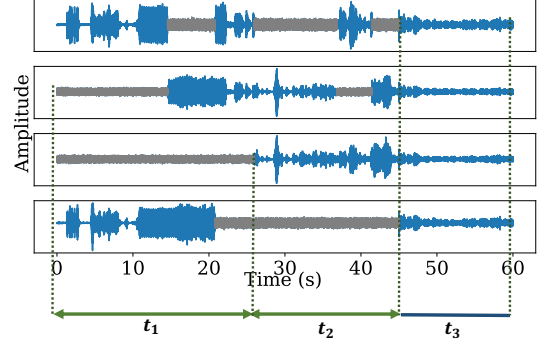

Figure 1: Example of raw audio from a four-channel microphone array. Each row represents one microphone; gray regions indicate missing data (75% MDP).

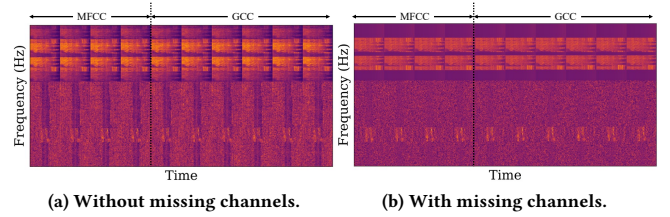**(a) Without missing channels.** **(b) With missing channels.**

Figure 2: Comparison of multi-channel features with and without missing data. Left: Clean MFCC and GCC; Right: Corrupted versions with missing channels.
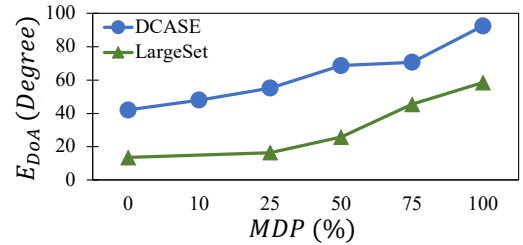

Figure 3: DoA error (E_{DoA}) increases as the Missing Data Percentage (MDP) rises. Results shown for a baseline SSL model evaluated on two datasets (details in Section 5.1) under varying levels of missing data.

Figure 1 shows an example with 75% MDP: during segments t_1 and t_2, at least one channel is missing, while t_3 has full data across all microphones. Since 75% of the total duration includes missing channels, the MDP is 75%. The temporal MDP is preferred over a channel-count-based MDP because it aligns with realistic, intermittent dropout patterns encountered in deployment, capturing both timing and duration of corruption, and enabling consistent evaluation across synthetic and real scenarios.
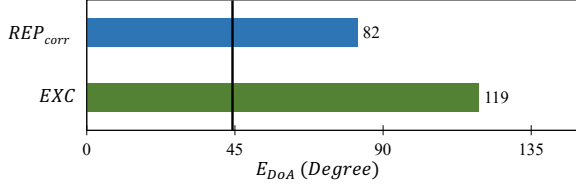
Figure 4: DoA error increases when using naive recovery methods: replacing with the most correlated channel (REP_{corr}) or excluding faulty channels (EXC).

Impact of Missing Channels on SSL. While SSL has been widely studied in both simulated and real-world settings [13, 44], the impact of missing channel data remains underexplored. Key features like Mel-frequency cepstral coefficients (MFCC) and Generalized Cross-Correlation (GCC) are highly sensitive to missing inputs, distorting the feature space and degrading DNN performance. As shown in Figure 2, missing channels significantly alter MFCC and GCC patterns. We quantify this by evaluating a standard SSL model [1, 42] on two datasets under increasing MDP levels; DoA error rises sharply (Figure 3), underscoring the need for more robust recovery methods.

Challenges in Replacing Corrupted Channels. Although nearby microphones are often correlated, naive approaches such as channel exclusion or correlated-channel replacement fail under dynamic dropout. We tested two such methods at 75% MDP: REP_{corr}, which replaces missing data with the most correlated working microphone's signal; and EXC, which excludes missing channels and retrains the DNN on only the active ones. As shown in Figure 4, both approaches increased DoA error by 46–63%. These results highlight the need for adaptive strategies like entropy-based recovery, which operate without prior knowledge of which channels are corrupted.

3 Problem Formulation and Mathematical Foundation

We begin by revisiting the concepts of **Full-Rank and Low-Rank Matrices**. A full-rank matrix F consists of linearly independent rows and columns, implying that all channels contribute distinct and complete information—critical for accurate sound source localization. In contrast, a low-rank matrix $\tilde{F}$ contains linearly dependent rows or columns, typically arising from missing or corrupted channel data, thereby degrading localization performance.

Let $X^{n \times T}$ represent the raw multichannel time-domain signal, where n denotes the number of channels and T is the time duration per channel. We define $X = \{x_1, x_2, \ldots, x_n\}$ as the set of input signals. A feature extraction function G maps this input to a feature domain as $F^{n \times D} = G(X)$, where D is the feature length per channel.

When all channels are present and functioning, F is full-rank. If one or more channels are missing or degraded, we denote the corrupted input as $\tilde{X}^{n' \times T}$ with $n' < n$, and its corresponding features as $\tilde{F}^{n' \times D} = G(\tilde{X})$. Our objective is to recover an estimate $\hat{F}$ that closely approximates the full-rank feature matrix F, i.e., $\hat{F} \approx F$.

To achieve this, we initially interpolate the missing features from $\tilde{F}$, yielding $\bar{\tilde{F}}$. However, this interpolation can inadvertently distort the original information, introducing additional error. To mitigate this, we reintroduce the relatively more reliable parts of $\tilde{F}$ as feedback into the interpolated features.

Because the reliability of individual elements in $\tilde{F}$ is unknown, we introduce a heuristic confidence mask $I(\tilde{F})$, which estimates the presence or absence of information. This mask is computed for each element (j, k) of $\tilde{F}$ as:

$$I_{j,k}(\tilde{F}) = \sigma(-\tilde{F}_{j,k} \log \tilde{F}_{j,k}) \tag{1}$$

Here, σ is the sigmoid function that bounds the confidence values between 0 and 1. This formulation is not derived from a formal probabilistic model, but rather serves as a heuristic measure of reliability. A value of 1 indicates high confidence in the information content of $\tilde{F}_{j,k}$, while 0 indicates complete absence, with intermediate values suggesting partial reliability.

Using this confidence mask, we refine the interpolated feature map as:

$$\hat{F} = \bar{\tilde{F}} + I(\tilde{F}) \otimes (\bar{\tilde{F}} - \tilde{F}) \tag{2}$$

This expression can be equivalently rewritten as:

$$\hat{F} = I(\tilde{F}) \otimes \tilde{F} + (1 - I(\tilde{F})) \otimes \bar{\tilde{F}} \tag{3}$$

In this formulation, $I(\tilde{F})$ acts as a confidence-driven selector: it retains elements from $\tilde{F}$ where the confidence is high and replaces uncertain elements with interpolated values from $\bar{\tilde{F}}$. Our goal throughout this paper is to estimate a full-rank feature matrix $\hat{F}$ from a corrupted, low-rank feature matrix $\tilde{F}$ using this reliability-guided reconstruction process.

4 LOCUS: LOcalization with Channel Uncertainty and Sporadic Energy

LOCUS estimates the full-rank feature matrix $\hat{F} \approx F$ from a corrupted low-rank matrix $\tilde{F}$, following the recovery formulation in Equation 3. As shown in Figure 5, *LOCUS* operates between the feature extractor and the DNN-based sound source localization (SSL) model. It consists of three key components: (1) Information-Weighted Focus (InFo), which estimates element-wise confidence score $I(\tilde{F})$; (2) Latent Feature Synthesizer (LaFS), which reconstructs a complete feature set $\bar{\tilde{F}}$ from $\tilde{F}$; and (3) Guided Replacement (GRep), which fuses $\tilde{F}$ and $\bar{\tilde{F}}$ using confidence masking $I(\tilde{F})$ to produce $\hat{F}$. This process is governed by Equation 3, and the resulting $\hat{F}$ is passed to the downstream SSL model for inference.

4.1 InFo: <u>In</u>formation-Weighted <u>Fo</u>cus

The purpose of **In**formation-Weighted **Fo**cus (**InFo**) module is to estimate the confidence score, $I(\tilde{F})$, for each element of the low-rank feature set $\tilde{F}$. The InFo module estimates the channel and spatial confidence score [35] by giving higher attention to elements with more available information. The input to InFo is the low-rank feature set $\tilde{F} \in \mathbb{R}^{d_{\tilde{X}}}$, where $d_{\tilde{X}}$ is the dimension of $\tilde{F}$. The left-hand side of Figure 5 shows the proposed early attention architecture, consisting of two main components.

The first component of InFo is the *channel confidence branch*, $f_\gamma(\tilde{F})$, parameterized with γ. This branch performs average pooling
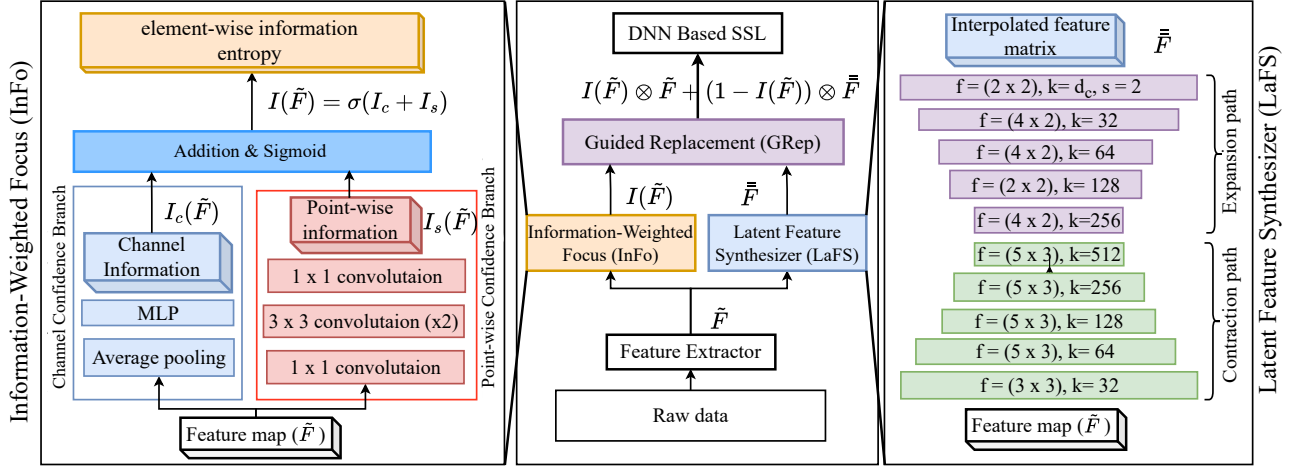
Figure 5: *LOCUS* consists of three key components for robust feature recovery under missing data. InFo estimates element-wise confidence score, $I(\tilde{F})$, from the low-rank feature map $\tilde{F}$ using both channel-wise and spatial feature branches. LaFS reconstructs a full-rank feature representation $\bar{\bar{F}}$ by modeling dependencies across available channels via a symmetric encoder-decoder structure. GRep combines the original and synthesized features using $I(\tilde{F})$ to produce the final recovered feature map $\hat{F}$.
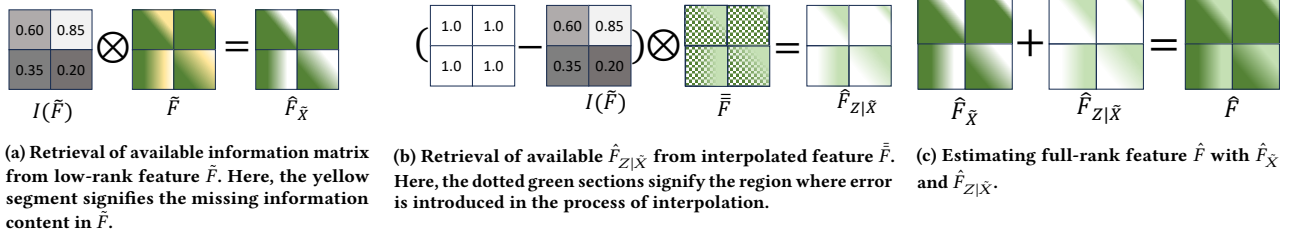

(a) Retrieval of available information matrix from low-rank feature $\tilde{F}$. Here, the yellow segment signifies the missing information content in $\tilde{F}$.

(b) Retrieval of available $\hat{F}_{Z|\tilde{X}}$ from interpolated feature $\bar{\bar{F}}$. Here, the dotted green sections signify the region where error is introduced in the process of interpolation.

(c) Estimating full-rank feature $\hat{F}$ with $\hat{F}_{\tilde{X}}$ and $\hat{F}_{Z|\tilde{X}}$.

Figure 6: Step-by-step workflow of Guided Replacement (GRep) to estimate the full-rank feature $\hat{F}$ from the low-rank feature $\tilde{F}$. Here confidence score $I(\tilde{F})$ acting as weight. When $I_{j,k}$ indicates low confidence in the original $F_{j,k}$, the system relies more on $\hat{F}j,k$; conversely, when $F_{j,k}$ appears reliable, it is preserved.

followed by a multi-layer perceptron to extract the channel confidence score. Average pooling aggregates information across all indices, enabling the model to learn more robust features while accurately depicting the overall strength of a feature. The multi-layer perceptron consists of three fully connected layers with batch normalization and rectified linear unit (ReLU) activation after the first two layers. This branch outputs a vector $I_c = f_\gamma(\tilde{F})$ that measures the global confidence score for each channel.

The second component of InFo is the *point-wise confidence branch*, $f_\lambda(\tilde{F})$, parameterized with λ. The input $\tilde{F}$ first passes through a convolution layer with a kernel size of (1×1) and a channel reduction factor of r. It then goes through n dilated convolution layers with a kernel size of (3×3), dilation value d, batch normalization, and ReLU activation. In our architecture, $n = 3$ and $d = 2$. The dilated convolution increases the receptive field, enabling us to leverage contextual information. Finally, this contextual information passes through another convolutional layer with a kernel size of (1×1) and outputs $I_s = f_\lambda(\tilde{F})$, which estimates the point-wise confidence score of the input feature.

Next, we broadcast I_s to $\mathbb{R}^{d_{\tilde{X}}}$ to match the dimension of $\tilde{F}$ and perform element-wise addition on I_c and I_s to estimate the information entropy. Finally, the *sigmoid* function, σ, maps the results between 0 and 1, where a value closer to 1 indicates higher information availability and vice versa. Therefore, the confidence score is defined as:

$$I(\tilde{F}) = \sigma(I_c + I_s) \qquad (4)$$

4.2 LaFS: **La**tent **F**eature **S**ynthesizer

The **La**tent **F**eature **S**ynthesizer (**LaFS**) estimates a full-rank feature set, $\bar{\bar{F}}$, from the available low-rank feature set $\tilde{F}$ by exploiting relations among the available channels. The right-hand side of Figure 5 shows the architecture of LaFS, which takes $\tilde{F}$ as input and provides the interpolated feature $\bar{\bar{F}}$ as output. We choose an auto-encoder for estimating $\bar{\bar{F}}$ as it can learn compressed intermediate representations. Auto-encoders can efficiently interpolate missing information by semantically mixing characteristics from data [5] and are used in latent variable generation [12, 20].

Table 1: Dataset Description.

Dataset	Duration	F_s(Hz)	No of samples	Train-Val-Test
DCASE	6 Hour	24000	600	400-100-100
LargeSet	50 hour	44100	30000	21000-3000-6000

LaFS contains two paths: (1) contraction and (2) expansion. The contraction path has 5 downsampling blocks, each consisting of a convolution layer (kernel size = 5×3), ReLU activation, and batch normalization. The output of the contraction path is the latent intermediate representation $F_{con} \in \mathbb{R}^{d_c}$, where d_c represents the intermediate feature dimension.

The expansion path passes the input F_{con} through 5 consecutive upsampling blocks, each consisting of transposed convolution with batch normalization and leaky-ReLU activation. The final convolution uses sigmoid activation without batch normalization. After each upsampling block, we halve the number of output feature filters, setting the last layer's output filter number to match the shape of the input feature map. The expansion path returns the interpolated feature matrix, $\bar{\bar{F}} = f_\phi(\tilde{F})$, where $\bar{\bar{F}} \in \mathbb{R}^{d_z}$.

4.3 GRep: Guided Replacement

The **G**uided **Rep**lacement (**GRep**) module recovers the full-rank feature $\hat{F}$ using Equation 3. Figure 6a depicts this retrieval process. First, we perform element-wise multiplication between the confidence mask $I(\tilde{F})$ and the low-rank feature set $\tilde{F}$ to retrieve the available information in the low-rank feature set $\hat{F}_{\tilde{X}}$.

Next, we find the missing information matrix $\hat{F}_{Z|\tilde{X}}$ from the interpolated feature $\bar{\bar{F}}$. Here, Z represents the missing data in $\tilde{X}$. As shown in Figure 6b, we take the complement of the confidence mask $I(\tilde{F})$ and multiply it element-wise with $\bar{\bar{F}}$ to find the missing information, disregarding regions (dotted darker green) where errors might have been introduced during interpolation.

Finally, GRep utilizes the available information matrix $\hat{F}_{\tilde{X}}$ and the missing information matrix $\hat{F}_{Z|\tilde{X}}$ to estimate the full-rank feature $\hat{F}$ from Equation 3 (Figure 6c).

5 Experimental Setup and Implementation Details

This section outlines the datasets, data preparation, implementation, baselines, and evaluation metrics used to assess *LOCUS*. We open-source our codebase and dataset [1].

5.1 Dataset

We evaluate *LOCUS* using two distinct multichannel sound source localization datasets, summarized in Table 1. Our evaluation covers both temporal (DCASE dataset, moving sources) and snapshot-based (LargeSet, static sources) DoA tasks, demonstrating *LOCUS*'s generalizability across varied temporal reasoning requirements.

Dataset 1: DCASE. The DCASE2021 Task 4 dataset [36] consists of 600 one-minute polyphonic recordings at 24 kHz, captured with a four-channel microphone array. The dataset includes up to four

[1]https://github.com/BASHLab/LOCUS

simultaneous sound sources and 12 distinct acoustic events, with Direction of Arrival (DoA) annotations for each source. It also features directional interference and multi-channel ambient noise, with SNR levels ranging from 6 to 30 dB. The dataset is split into 400 training and 200 evaluation samples.

Dataset 2: LargeSet. We created a 50-hour, six-channel dataset with 180,000 one-second recordings at 44.1 kHz across 10 diverse environments. This monophonic dataset avoids overlapping sound sources to isolate the impact of missing channels.

Simulation Details. We simulate a total of ten distinct acoustic environments with *PyRoomacoustics* [40] using the configuration parameters outlined in Table 2. Each environment represents a closed rectangular room with rigid boundaries, comprising a ceiling, floor, and four walls, where the material properties and layout may vary across environments to introduce diversity in reverberation and reflection characteristics. The dimensions of each room are defined in meters as $(l \times w \times h)$, representing the length, width, and height, respectively. Figure 7 and figure 8 depict the schematic of the simulation room and the microphone array geometry.

At the geometric center of each room, located at coordinate $\left(\frac{l}{2}, \frac{w}{2}, \frac{h}{2}\right)$, a spherical microphone array is placed. The array consists of six omnidirectional microphones uniformly distributed along the equator of the sphere. The radius of the microphone array is set to 4.25 cm, ensuring a compact configuration suitable for near-field sound localization.

To simulate sound sources within each environment, we use the *UrbanSound8K* dataset [39], to expand the diversity of acoustic content (e.g., engines, alarms, dog barks) and evaluate the generalization of our system beyond typical indoor scenes.. For each simulation instance, a single sound source location is randomly sampled from a continuous uniform distribution within the room volume. Specifically, the source coordinate (x, y, z) is drawn such that $x \in [0, l]$, $y \in [0, w]$, and $z \in [0, h]$, ensuring that the source is located strictly within the room boundaries. This coordinate is recorded as the ground truth for supervised learning tasks such as source localization or direction-of-arrival (DoA) estimation.

5.2 Data Preparation

Feature Extraction. We extract two key acoustic features: Mel-Frequency Cepstral Coefficients (MFCC) and Generalized Cross-Correlation (GCC) from raw audio using a 1024-point FFT with a 40 ms window and 20 ms hop length at 24 kHz and 44.1 kHz. MFCC captures magnitude-based audio information, while GCC captures time differences and phase shifts between channels, crucial for spatial localization. These features are combined into a matrix, $\tilde{F}$, which is input to the DNN-SSL pipeline.

Missing Channel Perturbation To simulate missing channels, we randomly select $m \leq \frac{n}{2}$ channels from n microphones, where m represents the missing channels. For each Missing Data Percentage (MDP), we replace $p\%$ of the audio with Gaussian noise (mean = 0, std = 1). We also simulate channel failures by randomly dropping entire channels for specific timeframes (100% MDP). These perturbations enable the model to handle real-world scenarios of microphone failure or signal distortion.
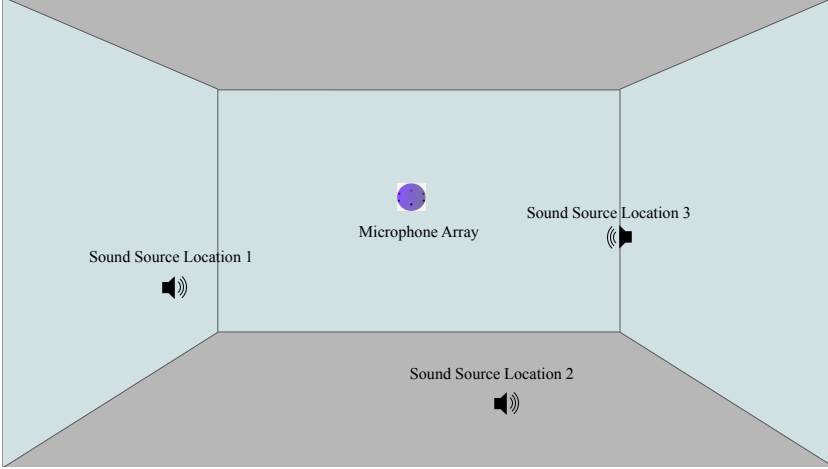
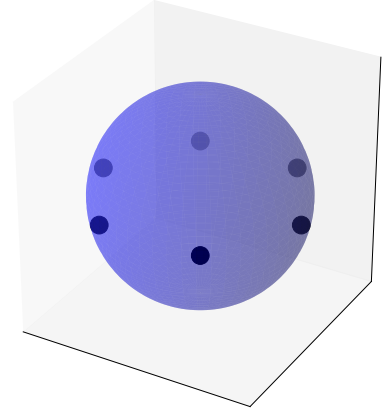**Figure 7: Simulation environment with the microphone array in the center.**

Figure 8: Microphone array

Table 2: LargeSet Simulation Environment

Dimension	Description
$10 \times 7.5 \times 3.5$	unpainted concrete - carpet, open cell foam - reverb chamber
$10 \times 7.5 \times 3.5$	plasterboard - cocos fibre roll - ceramic tiles
$15 \times 8 \times 5$	wooden lining - linoleum on concrete - lime wash
$15 \times 8 \times 5$	hard surface - carpet rubber - brickwork
$10 \times 7.5 \times 3.5$	plasterboard - carpet hairy - wooden lining
$10 \times 7.5 \times 3.5$	lime wash - carpet, closed cell foam - plasterboard
$15 \times 8 \times 5$	lime wash - felt - brick wall
$15 \times 8 \times 5$	lime wash - carpet cotton - concrete
$7.5 \times 4 \times 3$	plasterboard - carpet thin - rough concrete
$7.5 \times 4 \times 3$	concrete - carpet tufted - hard surface

5.3 Implementation Details

Training Pipeline of _LOCUS_. We train _LOCUS_ jointly with the DNN-based sound source localization (SSL) task. The SSL loss function, $\mathcal{L}_{DT}$, backpropagates through the network and the information branches (f_γ and f_λ) of InFo, enriching the features used for SSL.

The LaFS branch (f_ϕ) has a separate training objective to estimate the input features. Its loss function, $\mathcal{L}_{DCI}$, is the squared L_2 distance between the original and reconstructed feature matrices: $\mathcal{L}_{DCI} = ||F - \bar{\bar{F}}||^2$. This ensures that the reconstructed features closely match the original ones, promoting data consistency. We use two optimizers: one for training InFo and SSL with $\mathcal{L}_{DT}$, and another for training the LaFS branch with $\mathcal{L}_{DCI}$. This dual-optimizer approach ensures optimal performance for both components.

Sound Source Localization Pipeline. We implement two DNN-based SSL solutions for Direction of Arrival (DoA) estimation using multi-channel audio: SELDNet [1, 42] and SALSA [33], both utilizing two features (Section 5.2).

SELDNet is a Convolutional Recurrent Neural Network (CRNN) [7], combining CNNs for spatial feature extraction and RNNs for temporal dependencies. A fully connected (FC) layer estimates source locations in Cartesian coordinates, providing precise DoA estimates.

SALSA is an advanced SSL algorithm with a CRNN model based on the PANN ResNet22 [25] architecture. It includes a two-layer Bidirectional GRU and fully connected output layers, adapting to

input features by adjusting the number of input channels. During inference, sound classes above the detection threshold are considered active, and their corresponding DOAs are selected.

5.4 Evaluation Baseline

We evaluate _LOCUS_ by comparing it against four imputation methods and one deep interpolation technique, covering a range of strategies from simple imputation to advanced neural network models. This ensures a thorough comparison of _LOCUS_ 's effectiveness in handling missing data.

Imputation Methods. We benchmark _LOCUS_ against three signal processing and two deep learning imputation methods, including one designed for acoustic signals. Each method is evaluated on both pre-trained and retrained SSL networks, with retraining done using either corrupt or imputed data.

• _Mean Imputation (Mean)._ Replaces missing channel data with the mean of available channels [2], a simple and commonly used baseline method.

• _Hot Deck Imputation (HotDeck)._ Fills missing data using the most correlated channels [26], offering a context-aware alternative to mean imputation.

• _Probabilistic Imputation (Prob)._ Estimates missing data using probabilistic models [15], handling complexity and randomness in real-world datasets.

• _Deep Interpolation (AutoEnc)._ Uses autoencoders (AutoEnc) to interpolate missing data, comparing its performance in SSL to evaluate the effectiveness of DNN-based interpolation.

• _SoundSeive._ A state-of-the-art hierarchical interpolation method in the STFT domain [32], capturing spectral patterns for robust missing data imputation, providing a benchmark for _LOCUS_.

Evaluation Networks. We evaluate each method on (1) pretrained and (2) retrained SSL networks.

• _Pre-trained (PreTrain)._ The SSL network (e.g., SELDNet) is trained without missing channel data ($MDP = 0\%$) and tested on data with varying $p\%$ missing timeframes, assessing performance degradation.

Table 3: Computational complexity and overhead analysis of *LOCUS*, SELDNet, and SALSA.

Model	Number of Parameter	FLOPS	Execution Time	Power Consumption
SELDNet	595K	20B	1.04s	120.04J
LOCUS- SELDNet	3,686K	28B	1.08s	121.64J
SALSA	20,313M	169B	1.13s	123.96J
LOCUS- SALSA	20,560M	178B	1.19s	132.76J

• *Retrained (ReTrain).* The SSL network is retrained with corrupt or imputed data (*mean*, *HotDeck*, *SoundSeive*, and *AutoEnc*) at different MDP levels, evaluating the robustness of retraining on missing or imputed data.

We evaluate performance using the *Degree of Arrival Estimation Error (E_{DoA})*, which measures the angular difference between the estimated and true sound source locations. E_{DoA} quantifies localization accuracy, providing key insights into system's ability to determine the direction of incoming sounds, critical for spatial audio applications.

6 Results

This section begins by evaluating *LOCUS* on various This section evaluates *LOCUS* on various localization algorithms, comparing it to a state-of-the-art method to show current limitations. For a fair comparison, we compare against both pretrained and retrained versions of all baseline models under the same missing data conditions using the same training and evaluation splits as *LOCUS*. This ensures all models are equally exposed to corrupted input during learning. We analyze performance against baselines under varying missing information and conduct an ablation study to assess *LOCUS*'s components. Finally, we test *LOCUS* in a different sensor domain to show its generalizability. Section 7 further evaluates *LOCUS* in real-world environments.

6.1 Evaluation on Different Sound Source Localization Algorithms

We evaluate *LOCUS* on two sound source localization algorithms – SELDNet and SALSA. Figure 12 illustrates that with no missing information, SALSA and SELDNet achieve E_{DoA} values of 12.74 and 44.7 degrees, respectively. However, when MDP is 75% or 75% timeframe are missing channels, the E_{DoA} increases by 38.74 degrees for SALSA and 27.07 degrees for SELDNet. The introduction of *LOCUS* reduces these errors by 10.34 and 7.70 degrees, respectively.

Table 3 provides a complexity and overhead analysis for SALSA and SELDNet, both with and without *LOCUS*. It highlights that SALSA has 34093.87× more parameters and requires 8.45× more floating-point operations per second (FLOPS) compared to SELDNet. Figure 12 shows that, when combined with *LOCUS*, the simpler SELDNet (with only 595K parameters) achieves an E_{DoA} that is 14.01 degrees lower than SALSA. Moreover, *LOCUS* introduces only a minimal runtime overhead of $3.45 - 5.19\%$ and a slight increase in power consumption of $7.12 - 7.33\%$, which remains lower than SALSA alone.
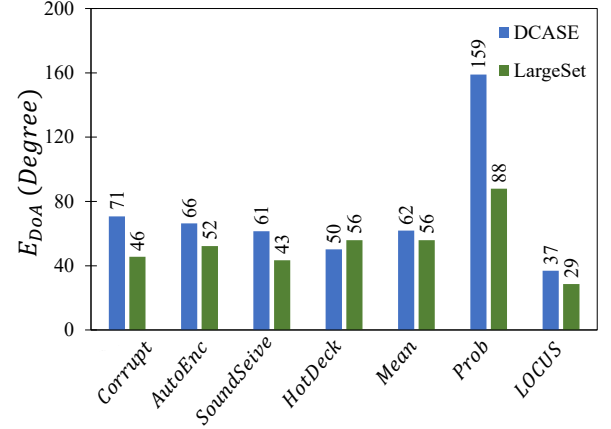

Figure 9: Comparison against various imputation methods evaluated using pre-trained models at MDP = 75%. *LOCUS* achieves the lowest E_{DoA} across both datasets, while the probabilistic imputation method shows the highest degradation, particularly on DCASE, highlighting the robustness of *LOCUS* under high data loss.
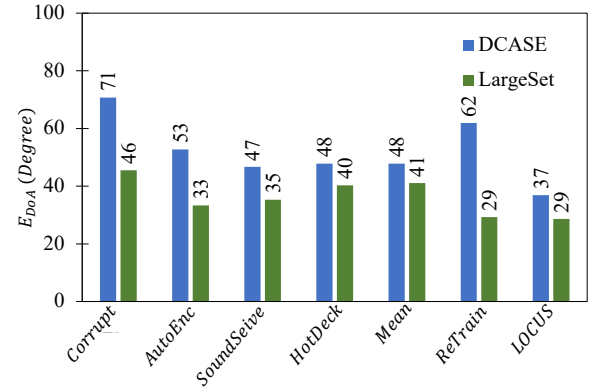

Figure 10: *LOCUS* achieves the lowest E_{DoA} across both datasets compared to *retrained* recovery methods with MDP=75%, indicating that simple retraining is insufficient to handle high missingness effectively.

6.2 Comparison with Baseline Algorithms

We analyze the performance of *LOCUS* against the baseline algorithms, as detailed in Section 5.4, using both the DCASE and LargeSet datasets.

Evaluation on Pre-trained SSL Network Figure 9 shows that without any imputation the *corrupt* data suffers from 70.70 and 45.52 degree E_{DoA} on DCASE and LargeSet, respectively, which is $47.89 - 36.95\%$ more than when all information is present. Three time-domain imputation techniques, *Mean*, *HotDeck*, and *Prob*, reduces this E_{DoA} to 50.14, 61.91, and 159.5 degree for DCASE and E_{DoA} 55.89, 55.68, and 87.88 degree for LargeSet. Imputation performs worse than the *Corrupt* baseline because it disrupts the inherent time-frequency structure of acoustic signals.
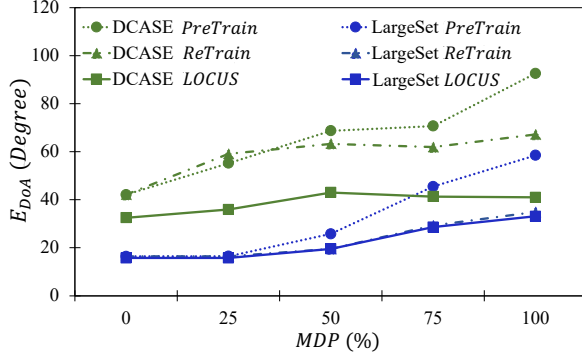
Figure 11: E_{DoA} versus *MDP%* for *PreTrain*ed SELDNet, *Re-Train*ed SELDNet, and *LOCUS* with SELDNet across DCASE and LargeSet. *LOCUS* maintains significantly lower error across all MDP levels, highlighting its superior robustness to missing data compared to conventional pretraining or retraining approaches.

AutoEnc has 66.30 and 52.30 degree E_{DoA} for DCASE and Large-Set, respectively which is better than *corrupt* but still $36.73 - 57.93\%$ more than when all information is present. The imputation targeted for Acoustic signals, *SoundSieve* improves performance over both *corrupt* and *AutoEnc* by reducing the E_{DoA} to 61.40 and 43.42 degree which is still $31.59 - 49.33\%$ lower than when all information is present.

On the contrary, *LOCUS*, recovers the missing features, and reduces E_{DoA} to 36.90 and 28.62 degree. Which is a $13.95 - 76.72\%$ improvement over all the baselines on DCASE. *LOCUS* also outperforms all the compared baselines $32.55 - 67.04\%$ on LargeSet.

Evaluation on Retrained SSL Network Figure 10 compares all retrained baselines with the proposed *LOCUS* model. Retraining the models with corrupted data improves their overall performance compared to pre-trained models. Although retrained models such as *HotDeck*, *Mean*, and *ReTrain* reduce E_{DoA}, their error rates remain $8.33 - 29.03\%$ higher than when full information is available on DCASE and $29.26 - 24.86\%$ on LargeSet. Further retrained models, *AutoEnc* and *SoundSeive*, achieve an additional reduction in E_{DoA} by $2.08 - 9.43\%$, yet they still perform $16.98 - 6.38\%$ worse than fully informed DCASE and $37.18 - 33.33\%$ worse on LargeSet. By contrast, *LOCUS* leverages mutual information across multiple channels, recovering performance by 15.19% on DCASE and 24.13% on LargeSet, respectively.

6.3 Effect of Missing Data Percentage

Figure 11 compares the E_{DoA} of *LOCUS*, *PreTrain*, and *ReTrain* when we have MDP = 25%, 50%, 75%, and 100%, which is corrupt data time-frames, on DCASE and LargeSet. *LOCUS* achieves $9.6 - 51.5$ and $0.64 - 25.4$ degrees lower E_{DoA} than *PreTrain* on DCASE and LargeSet, respectively. With a higher percentage of missing information, *PreTrain*'s performance degrades significantly, while *LOCUS* consistently retrieves missing information, maintaining variances of 4.39 and 7.89 degrees in E_{DoA} for DCASE and LargeSet. Compared to the retrained network with corrupt data (*ReTrain*), *LOCUS* achieves $9.6 - 26.1$ and $-0.05 - 1.78$ degrees lower E_{DoA} on DCASE
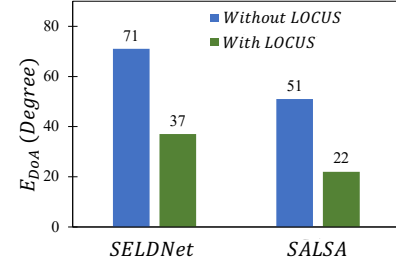

Figure 12: Comparison of *LOCUS* with SALSA at MDP=75%

and LargeSet, respectively. Both *PreTrain* and *ReTrain* show significant increases in E_{DoA} with higher MDP, as spatial relationships among input streams are crucial for localization tasks. The improvement is less pronounced for LargeSet compared to DCASE because LargeSet is monophonic with minimal MDP impact, whereas the polyphonic DCASE dataset experiences more corruption, highlighting *LOCUS*'s performance boost.
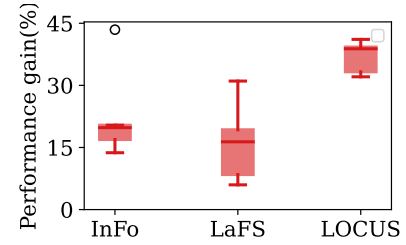

Figure 13: Improvement of E_{DoA} from $RTrain_p$ on DCASE with $MDP = \{10, 25, 50, 75, 100\}\%$.

6.4 Ablation Study

This section evaluates the contribution of each component of *LOCUS* using the DCASE dataset and SELDNet SSL network. Our ablation shows that this soft fusion using $I(\tilde{F})$ outperforms alternatives such as direct interpolation or naïve masking.

Effect of InFo. To understand the impact of InFo, we evaluate *LOCUS* without LaFS, leaving only InFo and the SSL network. Figure 13 shows that this configuration reduces the E_{DoA} by $22.85 \pm 10.57\%$ compared to retrained network with corrupt data. Adding InFo enhances performance by focusing attention on the elements of the feature matrix ($\tilde{F}$) that contain more information. However, without LaFS to estimate the values of the missing elements in the full-rank version, F, this setup performs $14.05 \pm 8.92\%$ worse than the full *LOCUS* implementation.

Effect of LaFS. Next, we investigate the effect of LaFS alone by removing InFo from *LOCUS*. This examination reveals the ability of an interpolator to recover complex missing information and the extent of additional errors introduced during interpolation. Figure 13 shows that LaFS alone reduces E_{DoA} by $16.22 \pm 8.87\%$ compared to retrained network with corrupt data. However, while LaFS estimates the full-rank features ($\bar{F}$), it unintentionally introduces errors in elements that are less affected by missing data. The inclusion of

Table 4: Environment Description for Real World Evaluation.

No	Dimension (ft)	Description
#1	$23.5 \times 46 \times 10$	A large room with no window and lots of furniture and electrical equipment.
#2	$18.5 \times 18 \times 10$	A medium size office space with multiple furniture and large glass windows.
#3	$11 \times 12.5 \times 10$	A small laboratory with multiple furniture and HVAC

InFo helps mitigate this by providing information entropy, which allows the replacement of these polluted components with their original elements, thereby reducing the overall error.

Effect of GRep. GRep in *LOCUS* combines the information entropy, $I(\tilde{F})$ from InFo with the interpolated full-rank feature $\bar{\bar{F}}$ from LaFS. As previously discussed, InFo estimates the available information matrix, $\hat{F}$ from $\hat{X}$, while LaFS interpolates to estimate the full-rank feature, $\bar{\bar{F}}$. For feature elements in $\hat{F}_{\tilde{X}}$ significantly impacted by missing channels, GRep replaces them with interpolated elements from $\bar{\bar{F}}$. Conversely, for elements less affected by missing channels, GRep preserves the original features from $\hat{F}_{\tilde{X}}$ using InFo. This method ensures a better estimation of $\hat{F}$, enabling *LOCUS* to outperform both InFo and LaFS alone by $16.22 \pm 8.87\%$ and $22.85 \pm 10.57\%$, respectively, as shown in Figure 13. These observations confirm that retraining the baseline model alone is insufficient and that using InFo or LaFS independently is inadequate. Thus, all three components of *LOCUS* are crucial for recovering corrupted information.

7 Real World Evaluation

This section evaluates *LOCUS* in three real-world environments with two energy scenarios, using ambient energy to power the microphone array.

Energy Harvesting Setup. We use an ETFE solar panel with a step-up regulator and Powercast harvester-transmitter system to harvest RF and solar energy across three environments. The solar panel, placed near a window, experiences intermittence from clouds and shadows, while the RF system faces disruptions from human traffic. Energy traces are collected from multiple MCU locations, each powered by different harvesters.

Acoustic Data Collection Setup. We use two MAX78000 microcontrollers to record from a four-channel microphone array, with each MCU connected to two microphones and powered by independent RF or solar harvesters. This distributed setup simulates real-world conditions where environmental factors, such as RF shadowing, multipath, or occlusion, lead to asynchronous power availability across nodes, resulting in partial channel dropouts. Each MCU logs data locally, and synchronization is achieved using periodic sync tones emitted by the sound source and captured by all microphones. This approach allows for offline alignment of data without the need for real-time synchronization. Due to I^{2}S constraints, we employ two MCUs to manage the microphone array, ensuring independent and efficient data collection across the system.
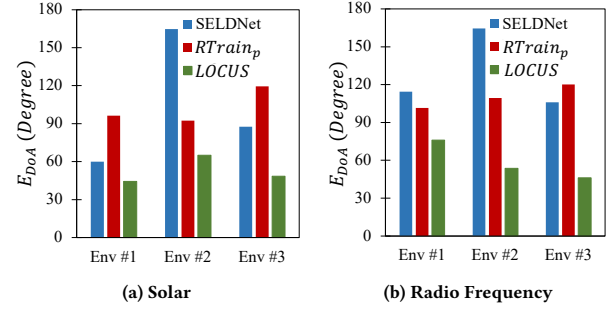

(a) Solar **(b) Radio Frequency**

Figure 14: *LOCUS* **with SELDNet recovers missing features more effectively and achieves lower** E_{DoA} **than** $RTrain_{75}$ **SELDNet for solar and RF harvested devices.**

We collect a polyphonic dataset with 1–3 concurrent audio classes, using 96 one-minute segments from seven classes in the NIGENS database [45]. The speaker is placed at 18 azimuthal angles (10° increments) across three environments (Table 4). Precise speaker locations are measured as ground truth, and background noise introduces natural variability. A total of 96 minutes of data are collected, with 72 for training and 24 for evaluation.

Results. Figure 14 shows that *LOCUS* outperforms both pre-trained and retrained SELDNet, reducing E_{DoA} by $25.87-60.55\%$ and $29.65-59.46\%$ in solar scenarios. Retrained SELDNet has higher E_{DoA} than the pre-trained model in environments 1 and 3 due to retraining on $MDP = 75\%$ data, which does not account for stochastic variations. In contrast, *LOCUS* effectively recovers missing information.

In the RF harvester scenario, where energy fluctuations are more frequent, Figure 14b shows that RF harvesting leads to more missing channels, causing higher errors for pre-trained SELDNet. Pre-trained and retrained SELDNet show $118-157\%$ and $33-160\%$ higher E_{DoA} than *LOCUS*. In contrast, *LOCUS* only shows $3.84 - 22.50\%$ more error than the no-missing-channels scenario, demonstrating its robustness. In contrast to solar harvesting, retrained SELDNet performs better than the pre-trained model in environment 1, suggesting that retraining on a fixed MDP does not adapt well to lower MDP conditions.

8 Related Work

Intermittent System. Early research in intermittent computing focused on data consistency under unreliable power, using checkpointing strategies to mitigate data loss during power interruptions. Systems like Mementos [37] implemented checkpointing but failed to address gaps in data during periods between checkpoints. Recent systems such as Greentooth [4] and FreeML [14] explore resilient architectures for energy-harvesting devices, while SONIC [18] improves DNN inference under intermittent power. Additionally, [30] shows audio sensing on intermittently-powered devices but is limited to short events (283 ms), insufficient for continuous data collected in real-world applications.

Missing Information Retrieval. Statistical methods such as mean imputation [2], hot deck [26], and multiple imputations [38] are used to handle missing data, but these methods are mainly designed

for single-channel data and struggle with long sequences of missing data [22]. In the deep learning domain, NLP [24] and CV [50] have explored missing data recovery, but these approaches mainly focus on text or image restoration. The closest work to our approach is SoundSieve [32], which addresses missing segments in single-channel audio due to energy intermittence, but it does not consider multi-channel systems. Other speech enhancement works, such as [31, 43], focus on recovering corrupted multi-channel data, but these are not directly applicable to sound source localization, which requires spatial data recovery. Recent multimodal work [27, 28] recover missing modalities, but ignores interrelated sensors in a single modality.

Sound Source Localization. Classic SSL methods, like MUSIC [19], independent component analysis [34], and sparse models [48], struggle with under-determined scenarios. Deep learning-based SSL has shown promise [16, 49], but fails in presence of missing or corrupted data.

Attention Mechanism. Attention mechanisms focus on critical information while ignoring irrelevant data [9]. In multi-channel input, attention is used to explore channel characteristics and estimate channel state information [17]. Although typically applied later in a network, early attention has shown substantial performance improvements [21], motivating the use of InFo in our approach.
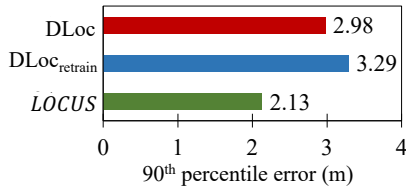
9 Discussions and Limitations

Figure 15: Result of WiFi localization at $MDP = 75\%$.

Beyond Audio and DoA. While *LOCUS* is designed for sound source localization, its feature recovery formulation is modality-agnostic. To demonstrate generalizability, we apply it to Wi-Fi CSI-based localization (**distance estimation**) using amplitude maps analogous to MFCCs. We evaluate *LOCUS* on the DLoc model [3] with a large-scale dataset covering two indoor environments and 16 CSI channels (4 access points × 4 antennas) under 75% missing data (MDP = 75%). We adopt DLoc's two-step architecture: an encoder processes CSI heatmaps, and two decoders predict spatial consistency and location. As shown in Figure 15, *LOCUS* reduces the 90th percentile localization error to 2.13 m 35.25% gain over retrained DLoc (3.29 m) and 25.52% over the original (2.98 m). This demonstrates *LOCUS* 's applicability beyond audio, including in RF sensing with incomplete data.

Offline Processing. *LOCUS* is designed for post-hoc analysis, where data is logged during deployment and processed offline, making it suitable for batteryless or energy-constrained systems where real-time inference is impractical. This allows us to employ complex recovery mechanisms without on-device computation costs,
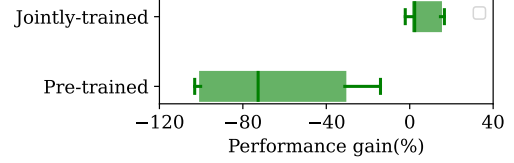

Figure 16: Impact of joint training on E_{DoA}.

though it limits its use in ultra-low-latency applications. Future work could explore lightweight, on-device approximations and integrate recovery-aware sensing strategies. Additionally, while we use sync tones for alignment, drift-aware synchronization could be explored for long-term distributed deployments.

Impact of Joint Training. *LOCUS* is jointly trained with the localization network to address feature distribution shifts caused by InFo and LaFS. As shown in Figure 16, joint training improves performance significantly. The recovery blocks are modular and can be trained separately for integration into existing systems, with future work exploring generalization strategies to reduce retraining requirements.

Generative Models for Information Retrieval. Generative models like VAEs and GANs are powerful but may struggle with reconstruction accuracy, particularly for discrete data. While they could replace the auto-encoder in LaFS, we find that a simpler auto-encoder performs comparably with less training overhead, making it a more efficient choice for multi-channel information retrieval in *LOCUS*.

LOCUS for Multi-Sensor Systems. While this paper focuses on multi-sensor systems like microphone arrays, *LOCUS* could be extended to multimodal systems (e.g., video, audio, text). For instance, *LOCUS* could recover missing audio data from video or text sources in a video conferencing setup, improving the overall quality.

Microphone Geometry Dependency. *LOCUS* relies on microphone-array geometry for localization and functions as a preprocessing layer that reconstructs reliable features before they are passed to downstream SSL architectures. This design allows us to support existing geometry-dependent methods without requiring modifications to their core structure. Future work will explore domain generalization techniques to address this limitation.

Large Real-World Localization Dataset. We created and released a 50-hour simulated dataset to train DNNs for sound source localization. However, simulated data may lack the complexity of real-world environments, and future work will focus on developing a comprehensive real-world dataset for sound source localization.

MDP incorporating number of missing channels. While incorporating the number of missing channels could add granularity, we excluded it since MDP is solely an evaluation metric, not used during training. Using a consistent MDP across varying missing-channel scenarios introduces a more challenging evaluation, better showcasing our method's robustness; jointly modeling temporal extent and channel count remains a promising direction for future work.

Image Recovery Limitations for Acoustic Features. While image pixel recovery methods are effective for images, they are unsuitable for frequency-domain features like GCC and MFCC,

which do not exhibit the same neighbor similarity. Thus, we cannot directly apply image recovery methods to restore missing acoustic channel data.

Handling Sparse Inputs. While *LOCUS* is not designed to operate on sparse inputs, it reconstructs missing features to restore dense representations before localization. Future work could explore architectures operating on sparse or partially observed features to reduce reconstruction overhead.

System-Level Exploration. A full exploration of communication overhead, runtime efficiency, and hardware acceleration is beyond the scope of this work. However, future efforts could focus on lightweight on-device approximations and energy-aware communication strategies for embedding *LOCUS* in fully deployed systems. To support on-device adaptation, we plan to explore a more comprehensive set of baselines and adopt efficient model architectures tailored for resource-constrained environments. Additionally, we plan to reduce feature dimensionality and apply model compression techniques such as pruning and quantization [6] to minimize memory and computational demands.

In this paper, we present *LOCUS*, an information retrieval-based sound source localization system designed to perform effectively despite missing or corrupted data. By leveraging the interdependence of data across channels, *LOCUS* recovers missing information that traditional algorithms, which require prior knowledge and struggle with high-complexity data, cannot handle. The InFo component quantifies both *how* much and *where* the data is corrupted, while LaFS interpolates the missing features. Our experiments demonstrate that *LOCUS* achieves up to 36.91% lower degree of arrival (DoA) error compared to retrained models using missing-channel data in the DCASE dataset. Additionally, in real-world scenarios with stochastically missing information, *LOCUS* shows performance gains of 25.87-59.46%, highlighting its robustness and effectiveness in real-world conditions.

Acknowledgment

References

[1] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. 2018. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing* 13, 1 (2018), 34–48.

[2] Paul D Allison. 2001. *Missing data.* Sage publications.

[3] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking.* 1–14.

[4] Simeon Babatunde, Arwa Alsubhi, Josiah Hester, and Jacob Sorber. 2024. Greentooth: Robust and Energy Efficient Wireless Networking for Batteryless Devices. *ACM Trans. Sen. Netw.* 20, 3, Article 66 (April 2024), 31 pages. https://doi.org/10.1145/3649221

[5] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. 2018. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543* (2018).

[6] Subrata Biswas, Mohammad Nur Hossain Khan, and Bashima Islam. 2025. QUADS: QUAntized Distillation Framework for Efficient Speech Language Understanding. *arXiv preprint arXiv:2505.14723* (2025).

[7] Yin Cao, Qiuqiang Kong, Turab Iqbal, Fengyan An, Wenwu Wang, and Mark D Plumbley. 2019. Polyphonic sound event detection and localization using a two-stage strategy. *arXiv preprint arXiv:1905.00268* (2019).

[8] Soumitro Chakrabarty and Emanuël AP Habets. 2017. Multi-speaker localization using convolutional neural network trained with noise. *arXiv preprint arXiv:1712.04276* (2017).

[9] Esther Luna Colombini, A da Silva Simoes, and CHC Ribeiro. 2014. *An attentional model for intelligent robotics agents.* Ph.D. Dissertation. Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil.

[10] Daniel de Godoy, Bashima Islam, Stephen Xia, Md Tamzeed Islam, Rishikanth Chandrasekaran, Yen-Chun Chen, Shahriar Nirjon, Peter R Kinget, and Xiaofan Jiang. 2018. Paws: A wearable acoustic system for pedestrian safety. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI).* IEEE, 237–248.

[11] Joseph Hector DiBiase. 2000. *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays.* Brown University.

[12] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. 2016. Adversarially learned inference. *arXiv preprint arXiv:1606.00704* (2016).

[13] Christine Evers, Heinrich W Löllmann, Heinrich Mellmann, Alexander Schmidt, Hendrik Barfuss, Patrick A Naylor, and Walter Kellermann. 2020. The LOCATA challenge: Acoustic source localization and tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 1620–1643.

[14] Pietro Farina, Subrata Biswas, Eren Yıldız, Khakim Akhunov, Saad Ahmed, Bashima Islam, and Kasım Sinan Yıldırım. 2024. Memory-efficient Energy-adaptive Inference of Pre-Trained Models on Batteryless Embedded Systems. *International Conference on Embedded Wireless Systems and Networks* (2024).

[15] Berihun Fekade, Taras Maksymyuk, Maryan Kyryk, and Minho Jo. 2018. Probabilistic Recovery of Incomplete Sensed Data in IoT. *IEEE Internet of Things Journal* 5, 4 (2018), 2282–2292.

[16] Eric L Ferguson, Stefan B Williams, and Craig T Jin. 2018. Sound source localization in a multipath environment using convolutional neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2386–2390.

[17] Jiabao Gao, Mu Hu, Caijun Zhong, Geoffrey Ye Li, and Zhaoyang Zhang. 2021. An attention-aided deep learning framework for massive MIMO channel estimation. *IEEE Transactions on Wireless Communications* 21, 3 (2021), 1823–1835.

[18] Graham Gobieski, Nathan Beckmann, and Brandon Lucia. 2018. Intermittent deep neural network inference. In *SysML Conference*, Vol. 48.

[19] Pooja Gupta and SP Kar. 2015. MUSIC and improved MUSIC algorithm to estimate direction of arrival. In *2015 International Conference on Communications and Signal Processing (ICCSP).* IEEE, 0757–0761.

[20] David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

[21] Amirhossein Hajavi and Ali Etemad. 2020. Knowing what to listen to: Early attention for deep speech representation learning. *arXiv preprint arXiv:2009.01822* (2020).

[22] Md Kamrul Hasan, Md Ashraful Alam, Shidhartho Roy, Aishwariya Dutta, Md Tasnim Jawad, and Sunanda Das. 2021. Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Informatics in Medicine Unlocked* 27 (2021), 100799.

[23] Bashima Islam Islam and Shahriar Nirjon. 2020. Zygarde: Time-Sensitive On-Device Deep Inference and Adaptation on Intermittently-Powered Systems.. In *ACM Proceedings on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT/UBICOMP '20).* ACM.

[24] Praveenkumar Katwe, Aditya Khamparia, Kali Prasad Vittala, and Ojas Srivastava. 2020. A Comparative Study of Text Classification and Missing Word Prediction Using BERT and ULMFiT. In *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020.* Springer, 493–502.

[25] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2020).

[26] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data.* Vol. 793. John Wiley & Sons.

[27] Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. 2022. Are Multimodal Transformers Robust to Missing Modality?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 18177–18186.

[28] Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. 2021. SMIL: Multimodal learning with severely missing modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 2302–2310.

[29] Amjad Yousef Majid, Patrick Schilder, and Koen Langendoen. 2020. Continuous Sensing on Intermittent Power. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* 181–192. https://doi.org/10.1109/IPSN48710.2020.00-36

[30] Amjad Yousef Majid, Patrick Schilder, and Koen Langendoen. 2020. Continuous sensing on intermittent power. In *19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* IEEE, 181–192.

[31] Payal Mohapatra, Shamika Likhite, Subrata Biswas, Bashima Islam, and Qi Zhu. 2024. Missingness-resilient Video-enhanced Multimodal Disfluency Detection.

In *Interspeech 2024*. 5093–5097. https://doi.org/10.21437/Interspeech.2024-1458

[32] Mahathir Monjur, Yubo Luo, Zhenyu Wang, and Shahriar Nirjon. 2023. Sound-Sieve: Seconds-Long Audio Event Recognition on Intermittently-Powered Systems. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 28–41.

[33] Thi Ngoc Tho Nguyen, Karn N. Watcharasupat, Ngoc Khanh Nguyen, Douglas L. Jones, and Woon-Seng Gan. 2022. SALSA: Spatial Cue-Augmented Log-Spectrogram Features for Polyphonic Sound Event Localization and Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2022), 1749–1762.

[34] Tahereh Noohi, Nicolas Epain, and Craig T Jin. 2013. Direction of arrival estimation for spherical microphone arrays by combination of independent component analysis and sparse recovery. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 346–349.

[35] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. 2018. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514* (2018).

[36] Archontis Politis, Sharath Adavanne, Daniel Krause, Antoine Deleforge, Prerak Srivastava, and Tuomas Virtanen. 2021. A Dataset of Dynamic Reverberant Sound Scenes with Directional Interferers for Sound Event Localization and Detection. In *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*. Barcelona, Spain, 125–129. https://dcase.community/workshop2021/proceedings

[37] Benjamin Ransford, Jacob Sorber, and Kevin Fu. 2011. Mementos: system support for long-running computation on RFID-scale devices. In *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems* (Newport Beach, California, USA) *(ASPLOS XVI)*. Association for Computing Machinery, New York, NY, USA, 159–170.

[38] Donald B Rubin. 1996. Multiple imputation after 18+ years. *Journal of the American statistical Association* 91, 434 (1996), 473–489.

[39] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*. 1041–1044.

[40] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. 2018. Pyroomacoustics: A python package for audio room simulation and array processing algorithms.

[41] Ralph Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation* (1986).

[42] Kazuki Shimada, Yuichiro Koyama, Naoya Takahashi, Shusuke Takahashi, and Yuki Mitsufuji. 2021. ACCDOA: Activity-coupled cartesian direction of arrival representation for sound event localization and detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 915–919.

[43] Hassan Taherian, Sefik Emre Eskimez, Takuya Yoshioka, Huaming Wang, Zhuo Chen, and Xuedong Huang. 2022. One model to enhance them all: array geometry agnostic multi-channel personalized speech enhancement. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 271–275.

[44] Tan-Hsu Tan, Yu-Tang Lin, Yang-Lang Chang, and Mohammad Alkhaleefah. 2021. Sound source localization using a convolutional neural network and regression model. *Sensors* 21, 23 (2021), 8031.

[45] Ivo Trowitzsch, Jalil Taghia, Youssef Kashef, and Klaus Obermayer. 2019. *NIGENS general sound events database*.

[46] Jorge R Vergara and Pablo A Estévez. 2014. A review of feature selection methods based on mutual information. *Neural computing and applications* 24, 1 (2014), 175–186.

[47] Nelson Yalta, Kazuhiro Nakadai, and Tetsuya Ogata. 2017. Sound source localization using deep learning models. *Journal of Robotics and Mechatronics* 29, 1 (2017), 37–48.

[48] Zai Yang, Jian Li, Petre Stoica, and Lihua Xie. 2018. Sparse methods for direction-of-arrival estimation. In *Academic Press Library in Signal Processing, Volume 7*. Elsevier, 509–581.

[49] Mariam Yiwere and Eun Joo Rhee. 2017. Distance estimation and localization of sound sources in reverberant conditions using deep neural networks. *Int. J. Appl. Eng. Res* 12, 22 (2017), 12384–12389.

[50] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. 2017. Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 351–355.