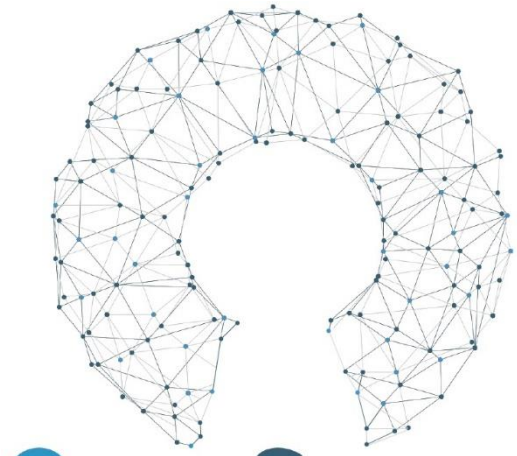

www.openswarm.eu

Call: HORIZON-CL4-2022-DATA-01

Type of action: RIA

Grant agreement: 101093046

Deliverable n°3.3 : Analysis of Decentralized GNN Knowledge Distribution and Privacy-Preservation

Work Package n°3: Collaborative energy-aware AI

imec

WP Lead: imec

This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046.

Funded by the European Union

Document information	
Author(s)	Matthias Hutsebaut-Buysse, Thomas Avé, Wei Wei, Kevin Mets
Reviewers	Markus Sauer (internal)
Submission date	31-Oct-2024
Due date	31-Oct-2024
Type	Public
Dissemination level	

Document history			
Date	Version	Author(s)	Comments
28-Jun-2024	01	Matthias Hutsebaut-Buysse, Wei Wei	Preview
23-Oct-2024	02	Matthias Hutsebaut-Buysse, Wei Wei	Deliverable

1. Table of contents

1. TABLE OF CONTENTS 2

2. EXECUTIVE SUMMARY 5

3. INTRODUCTION 6

4. PAPERS 7

5. KPIS 7

6. TRL LEVEL 9

7. BACKGROUND 9

7.1. Graph-structured data 9

7.2. Graph neural network 9

7.3. Distributed GNN 10

7.4. Decentralized learning with GNN 10

7.4.1. GNN-assisted decentralized federated learning 11

7.4.2. Decentralized knowledge distillation 11

7.5. Sensor fusion 11

7.5.1. Sensor Fusion with audio modality 13

7.5.2. Feature-level fusion with audio modality 13

8. RULE-BASED SWARM INTELLIGENCE 15

8.1. Implementation of baselines 15

8.1.1. Majority voting 15

8.1.2. Maximum confidence 15

8.1.3. Sum of confidences 16

8.2. Study on SINS datasets 16

8.3. Building of a custom dataset 17

8.3.1. Motivation **17**

8.3.2. Procedure **18**

8.4. Results **19**

9. DEEP-LEARNING-BASED SWARM INTELLIGENCE **21**

9.1. Implementation of baseline **21**

9.1.1. Classifier stacking **21**

9.1.2. CNN-based feature-level fusion **22**

9.2. Feature-level fusion in wireless acoustic sensor networks with graph attention network for classification of domestic activities **23**

9.2.1. Methodology **27**

PROBLEM DESCRIPTION 27

FROM WASN TO GRAPH 28

GRAPH ATTENTION NETWORK 28

9.2.2. Framework **30**

ON-DEVICE FEATURE EXTRACTOR 30

MESSAGE CONDENSATION 31

GNN-BASED SENSOR FUSION 32

GRAPH-LEVEL PREDICTION 32

9.2.3. Additional Baselines **33**

9.2.4. Experiments setting **33**

9.2.5. Results **35**

9.2.6. Ablation study **38**

9.2.7. Conclusion **39**

9.3. Graph neural network for underwater sound source localization **41**

9.3.1. Related work **41**

9.3.2. Dataset **42**

DATASET OF LOUISE ET AL. [69] 42

SHIPSEAR DATASET 43

DEEPSHIP DATASET 43

PASSIVE ACOUSTIC MONITORING DATASETS 44

SIMULATED DATASETS 45

9.3.3. Preliminary results **45**

QUANTITATIVE RESULTS 45

QUALITATIVE RESULTS 47

9.3.4. Conclusion **48**

10. CONCLUSIONS 48

11. REFERENCES 50

2. Executive Summary

Nodes in a swarm are most often equipped with one or multiple sensors. In order to make decisions the sensor of a single node can be utilized locally. However, a lot of value can be obtained from combining the observations from multiple sensors. E.g., if nodes are deployed in physically spread-out locations, a larger area can be processed then what would be possible when utilizing a single sensor. Additionally, if a node would become unavailable the whole swarm could still be able to carry out its functions through redundancy. However, combining sensor data in a dynamic and energy-aware manner is no trivial task. AI-techniques such as Graph Neural Networks (GNNs) have been deemed effective in combined such linked data in other areas.

So, within this document we formulate the hypothesis that GNN-based methods can be efficiently utilized in swarms to efficiently process distributed high-dimensional observations. We outline the work that we have done in order to make an AI-enabled swarm of sensor nodes possible, focussing on being low-power on the nodes themselves through limiting communication overhead.

We conducted a survey to research existing benchmarking tasks and performance of SOTA methods in acoustic activity classification tasks. However, as the well-studied existing benchmarking tasks did not fully reflex the scenario of distributed “Ocean Noise Pollution Monitoring” we wanted to support, to demonstrate the AI-based novel methods that we have been developed we first constructed an acoustic activity simulator that allows us to simulate a swarm of microphones (a wireless acoustic sensor network or WASN) placed inside scans of existing buildings. We utilized this simulator and the related SINS benchmark in order to validate our hypothesis that utilizing graph neural networks in this context in beneficial to the overall classification accuracy for acoustic activity classifications tasks. Additionally, we also proposed an extension of our method that is able to estimate the location of a sound source.

The novel developed algorithms can be utilized as reference implementations in the "OpenSwarm implementation" and within the proof of concepts (especially PoC3: Ocean Noise Pollution Monitoring).

3. Introduction

Low-power, on-field device learning methods make it possible to seamlessly scale to collaborative networks of numerous devices. Each of the field devices will be capable of processing its own sensed environment and can perform the necessary analysis, predictions, etc., for its area of interest, independently.

There is an interesting connection to be made between swarms of devices and graph neural networks (GNN). For example, each field device in the swarm could form a node in a distributed graph. In this way, the swarm becomes itself an inference and learning unit. Feature vectors obtained from each node -- i.e., field devices -- can be communicated to neighbouring nodes. The visual is that of these feature vectors forming the edges in the GNN, with each node/field device forming a vertex. By performing the analysis at the node level, only the results (features) need to be communicated, instead of the raw data collected from each field device's sensor(s). This reduces the bandwidth and latency requirements, as only minimal numerical vectors need to be transmitted across the graph.

A further outcome is that of privacy preservation, as the original data is never shared between devices.

In the remained of this document we will first introduce the concepts that are used throughout the rest of the document in Section 7. Then we describe the baseline experiments that were conducted through utilizing a rule-based approach in Section 8. We improve upon these methods through utilizing deep-learning-based methods which are outlined in Section 9.

4. Publications

Some results which are outlined in this report have been presented at the following conference:

- Feature-level fusion in wireless acoustic sensor networks with Graph Attention Network for classification of domestic activities. Wei Wei, Matthias Hutsebaut-Buysse, Thomas Avé, Tom De Schepper, Kevin Mets. DAFUSAI ECAI24 workshop, Santiago de Compostela, Spain, 19-24 October 2024

Additionally an improved version of the proposed method has been described in the following publication:

- Flexible and Efficient Feature-level Fusion with Wireless Acoustic Sensors using Graph Attention Networks. Wei Wei, Matthias Hutsebaut-Buysse, Thomas Avé, Tom De Schepper, Kevin Mets IEEE Sensors Journal. (under review)

5. KPIs

To measure the outcome of the work presented in this document a KPI was set on the **target number of collaborative AI nodes in the swarm**. The target number of nodes has been set to be at least 32.

With this goal in mind, we developed a graph neural network (GNN) based sensor fusion framework that enables the nodes in the swarm to collaborate by sharing their data to a centralized fusion center. This framework can accept a variable amount of input, our framework will be able to predict with a swarm of size 32 and higher. The only difference between a small and a large swarm for our framework will be the time spent for computation.

To examine the impact of swarm size versus the required computation time, we conducted a scalability experiment with an increasing swarm size. The results is visualized in Figure 1.

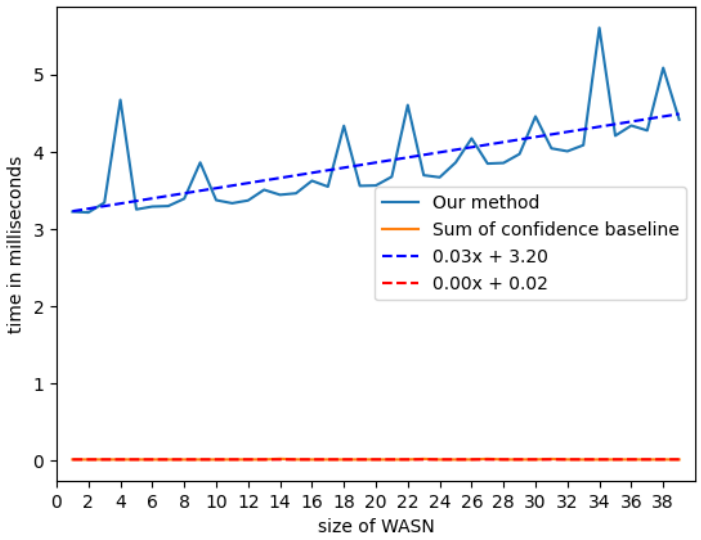

Figure 1: The result of the scalability study for the developed GNN-based sensor fusion framework. We observe a linear correlation between the computation time and the size of the WASN for our framework. However, the impact of the large WASN is limited, each additional node increases the computation time by approximately 0.05 milliseconds.

The computation time of our framework has a linear correlation with the size of the Wireless Acoustic Sensor Network (WASN), while the sum of confidence baseline has a constant computation time. However, we observe that the impact of the size of the WASN to the computation time of our framework is limited. A linear regression method is used to find the best fitting function that expresses the evolution trend of the computation time. We see that the coefficient of the function is 0.05, denoting that each additional node in the swarm only adds approximately 0.05 millisecond to the run-time of our framework. Thus, we can conclude that our framework will successfully scale to a use case with 32 nodes in the swarm.

6. TRL level

We started the work package 3.3 at TRL2 (technology concept formulated). Currently we are at TRL3 (experimental proof of concept). Through further efforts we aim to increase the TRL-level. The OpenSwarm project in general is aiming for a TRL leap from 2 to 5. In order to do this we have kept the proposed proof of concepts in mind while developing our method. This should already contribute in the next increases in TRL-level.

7. Background

7.1. Graph-structured data

In graph theory, an undirected graph $G = (V, E, A)$ consists of the node set V with N nodes $v_i \in V$ and the edge set E with E pairs of nodes $(v_i, v_j) \in E$. The graph adjacency matrix A is a matrix of size $|V| \times |V|$ where $A_{ij} = 1 \leftrightarrow (v_i, v_j) \in E$.

Each node v_i has a feature vector $h_i \in R^F$, which we denote as the node feature or the node embedding. A fully connected or a complete graph refers to the graph where each node has an edge to every other node.

7.2. Graph neural network

GNNs are neural networks designed to deal with graph-structured data. A graph neural network consists of several layers of graph convolution or graph recurrent operations [1]. We focus on the graph convolution operations in this report. A graph convolution can either be spectral-based or spatial-based.

Spectral-based graph convolutions are theoretically based on graph signal processing [2]. They usually require to perform operations on the graph adjacency matrix A, such as the eigenvalue decomposition [3]. This operation requires the graph adjacency to

remain unchanged between train and test time. In the case of OpenSwarm, we want to allow dynamic networks of sensors, so this will not be the case, thus, we focus on a spatial-based graph convolution.

A spatial-based graph convolution aggregates the information only from the neighborhood of each node [1]. To obtain the new node feature h_n^{t+1} for a node n, a spatial-based graph convolution aggregates the features of the nodes that are in the first-order neighborhood $\mathcal{N}$ of n with an aggregation function AGG as described below:

$$h_n^{t+1} = AGG(\{h_k^t, \forall k \in \mathcal{N}_n\})$$

thus, only the partial graph structure is required for the convolution operation of each node. However, the aggregation function AGG must be order-invariant and it must accept a variable number of inputs. An example of such an aggregator is the MAX operation. To increase the expressiveness of the GNN, non-linear activation functions and deep learning-based transformations are usually performed before or after the aggregation process.

7.3. Distributed GNN

The research area of distributed GNNs focuses on building frameworks to optimize the learning speed of GNNs by utilizing multiple GPUs across different locations. The focus lies on distributed data storage, parallelizable operations and training a large GNN across multiple resource clusters.

Recent works include AliGraph [4], GraphTheta [5], DistGNN [6], DistDGL [7]. The objective of these works is to optimize the training time of a GNN on a large graph that is already collected and stored, which is out of scope of the OpenSwarm project.

7.4. Decentralized learning with GNN

In contrast to distributed learning, decentralized learning aims to train a model across multiple devices without assuming that the data is stored on a central server. In particular, decentralized federated learning trains a model across multiple data owners while limiting the exchange of raw data. This is more in line with the overall objective of the OpenSwarm project.

7.4.1. GNN-assisted decentralized federated learning

The survey by Rui Liu, et al. [8] defines the concept of GNN-assisted federated learning. In this setting, each data owner in the federated learning group can be seen as a node in a graph neural network. They can either all communicate to a centralized server which aggregates their information or communicate with neighboring data owners to enable decentralized federated learning.

This concept aligns highly with the objective of task 3.3 as we aim to develop mechanisms for knowledge sharing between intelligent nodes, without sharing raw data between the nodes.

Related literature in GNN-assisted decentralized federated learning include SpreadGNN [9], D-FedGNN [10], and WD-GNN [11].

7.4.2. Decentralized knowledge distillation

Ilai Bistritz et al. [12] proposed that the devices can learn from each other via sharing prediction results on a reference dataset to each other. This method satisfies the requirement for task 3.3 too.

In their work, the devices share softmax-function outputs, e.g. probability of the classification in order to perform knowledge distillation on neighboring devices to achieve collaborative training between the edge devices. In this framework, the devices don't need to share raw data with each other, and the architecture of the machine learning model deployed on the devices can be different. It is an interesting alternative to the decentralized federated learning concept to achieve the objective of task 3.3.

7.5. Sensor fusion

While the aforementioned research can be used to implement a swarm intelligence, it is typically assumed that the edge device has ample energy for the data processing and transmission. In wireless sensor fusion, the energy consumption of the edge device is also an important constraint [13], and there are existing research works that use sensor fusion to aggregate acoustic signals from edge devices [13], [14], [15], [16]. We note that the processing of acoustic signals is important for multiple proof of concepts proposed

in the OpenSwarm project. Thus, realizing swarm intelligence with sensor fusion for wireless acoustic sensors will an ideal reference use case.

Sensor fusion is the task of aggregating data or knowledge from multiple sensors. The term 'data fusion' or 'feature fusion' is also used by previous studies for the task of multimodal information aggregation [17] and the aggregation of results obtained by multiple machine learning models on one data source [18], [19], [20], [21]. Our work focuses on the aggregation of data from multiple sensors for the audio modality via a centralized fusion center. We discuss the related works in this section.

According to the model proposed by Dasarathy et al. [22]. There are three levels of sensor fusion:

1. Data-level fusion
2. Feature-level fusion
3. Decision-level fusion

Traditional data-level fusion methods includes Bayesian fusion, evidential belief reasoning and rough-set based fusions [23], [24]. There are also methods that use machine learning models such as support vector machine [25] and neural networks [26]. These methods require the sensor to transmit the raw data they observe to the remote fusion center. However, transmitting raw data over a wireless network requires the most energy compared to the other two levels of sensor fusion. As the communication layer dominates the energy consumption of the sensor [27], the data-level fusion yields the worst battery life and is less favorable.

As the processing of observations locally has become possible with recent hardware advancements (smart sensors), feature-level fusion and decision-level fusion have become more interesting approaches for downstream applications that require a low power consumption. Compared to data-level fusion, feature- and decision-level fusion with deep neural networks are explored in the application of autonomous driving [28], [29], [30], [31], machine condition monitoring [32] and smart city management and monitoring [33], [34]. The focus of the previous studies is divided between the fusion of

multimodal sensor data and the sensor data from different locations. The research in multimodal sensor fusion is distinctively different compared to sensor fusion with a wireless sensor network. The former focuses on exploring cross-modal information [35], while the latter focuses on increasing the spatial resolution within one modality [13]. Our work focuses on the latter case, and specifically on the audio modality.

7.5.1. Sensor Fusion with audio modality

For sensor fusion with an audio modality, event classification is widely used as a benchmarking task [36]. Previous datasets include DIRHA [37], UPC-TALP [38], SINS [39] and Sweethome [40]. The latter two datasets are designed for the classification of domestic activities with acoustic data. Giannoulis et al. [41] explored different fusion strategies to fuse information from multichannel audio data. They proposed to use a plain delay-and-sum beamformer to combine signals from multiple channels. Additionally, they computed the time-difference of-arrival between channel signals as an additional feature. Finally, they also experimented with different decision-level fusion strategies, such as sum of confidences and majority voting. Later, Martín-Morató et al. [42] evaluated different data- and decision-level fusion strategies in different acoustic scenarios. Dekkers et al. [13] focused on the energy consumption of different decision-level fusion strategies, and proposed a dynamic sensor activation strategy to reduce the overall energy cost.

Apart from the abovementioned works, Kürby et al. [15] proposed the classifier stacking decision-level fusion strategy. This strategy learns a random forest classifier to aggregate the decisions of the audio sensors into one final decision. This strategy outperforms other decision-level fusion strategies studied in the abovementioned works. The same result is confirmed by Grzeszick et al. [14]. However, a limitation of this strategy is that the learned random forest classifier is not order invariant and cannot deal with additional sensors or sensor failure.

7.5.2. Feature-level fusion with audio modality

In a feature-level fusion method, each sensor transmits a latent feature vector to the centralized fusion center, and the fusion center aggregates all the information to produce a final decision. Compared to the decision-level fusion methods, the latent

feature vectors convey richer information, which means that a feature-level fusion method will usually yield a better performance. Feature-level fusion with audio data has been used in the field of fault diagnosis. Xu et al. [43] transformed the multichannel audio data collected by the sensors to a multichannel image. Each audio channel corresponds to a color-channel of the image. The raw observation data points correspond to the pixels in the image. A shared CNN was then used to transform the data from all the sensors to latent features, and the features were concatenated and used for prediction. Li et al. [16] used similar approaches where the audio data were transformed into images. However, they first preprocessed the raw observations using a fast Fourier transform filtering. A shared CNN is used to extract latent features, and the concatenation operation was used to aggregate the latent features from the different sensors. Finally, Zhang et al. [44] is the most similar to our proposed method. They use a graph attention network (GAT) for sensor fusion with multiple sound segments collected by different sensors from different timestamps. However, they do not take the core challenges in the wireless setting into account. They construct a graph for each sensor separately, then use a graph attention layer (GAL) to extract latent features. The extracted features are concatenated and used as new node features, which they feed into the second GAL for classification. The concatenation process after the first GAL makes their proposed framework unable to handle a variable number of sensors. For all the prior feature-level fusion methods, if one of the sensors fails to transmit the data, the concatenation process will output a shorter output, which is an invalid input for the following layers. In our framework, a transmission failure will result in the removal of a node in the graph, thus, the length of the node features in the graph is always the same. This allows our framework to perform inference regardless of the number of sensors.

8. Rule-based swarm intelligence

8.1. Implementation of baselines

An intuitive way to achieve swarm intelligence is by implementing rules that decides the outcome based on the decisions from multiple agents in the swarm. This corresponds to the decision-level fusion methods in literature [13], [14], [15]. Several baselines from this category were implemented and they are explained in this section.

8.1.1. Majority voting

This is a straightforward fusion strategy that takes the class chosen by most of the sensors as the final decision. If we denote the vector of confidences of the classes $y \in Y$ of a sensor k for an observation x as $c(x)_k$, and the class with the highest confidence in $c(x)_k$ as y'_k. This fusion strategy can be described as:

$$y'_{final} = \underset{y}{argmax} \sum_{k} 1 * \delta(y'_k, y)$$

Here, δ denotes the Kronecker delta function that is only true when y'_k equals to the y which we are looking for. This fusion method assumes that the most sensors will be able to classify the scene correctly. Each sensor only needs to send y'_k to the fusion center, the confidences $c(x)_k$ do not need to be sent back. Thus, it has the lowest communication cost.

8.1.2. Maximum confidence

This fusion strategy takes the class with the highest confidence amongst different sensors as the final decision, it can be described as:

$$y'_{final} = \underset{y}{argmax} \ \underset{k}{max} \ c(x)_k$$

Here, the sensor with the highest confidence is chosen, then the corresponding class which has the high confidence is chosen as the final class, regardless of the decision of other sensors. This fusion method requires each sensor to send back both the label and

the confidence score of the class with the highest confidence, i.e., the maximum of the output of the classification layer. This incurs a higher communication cost compared to the last method.

8.1.3. Sum of confidences

This fusion strategy sums the confidences of all classes over all sensors, and take the class with the highest confidence after summation, it can be described as:

$$y'_{final} = \underset{y}{argmax} \sum_{k} c(x)_k$$

Here, the sensors must send the confidence score of all the classes to the remote fusion center. This method has a higher communication cost compared to the `highest confidence' fusion strategy.

8.2. Study on SINS datasets

To evaluate the performance of the rule-based swarm intelligence methods, the task of acoustic based domestic activity classification was chosen. It is closely related to the OpenSwarm proof-of-concept for the ocean noise pollution monitoring, where the task is to identify and count the boats inside a certain perimeter and monitor their speed.

The SINS dataset [39] is widely used for this task. It consists of continuous audio recordings of a vacation home over a period of one week. It was collected using a network of 13 microphone arrays distributed over multiple rooms. Each microphone array has 4 linearly arranged microphones. We refer to the original publication for the positions of the microphone arrays [39].

The recordings and event labels of the living room were used for our experiments. There are 8 microphone arrays deployed in the living room, and the data of 7 microphone arrays were made public (microphones with ID {1,2,3,4,6,7,8}). We used the event intervals, and the recording timestamps provided by Dekkers et al. [5] to synchronize and label the recordings between the different sensors. We clipped the recordings into 10 second segments with a sampling frequency of 16kHz, and calculated the mean of the different microphones in the microphone array to obtain a

mono-channel audio signal. During this process, we noticed that there is a heavy class-imbalance in the SINS dataset. As an example, the "absence" class has much more recordings compared to other classes. To limit the class imbalance, we set an upper limit of 300 clips per class per listener.

After the synchronization and segmentation, we applied a Short-Time Fourier Transform (STFT) on the audio segments. A window length of 400ms with hop length of 200ms is used to create frames within the audio segment. The STFT magnitude of the frame is forwarded to a Mel-scale filterbank with 64 bands and a frequency range of 0 to 8000 Hz followed by a logarithmic transformation at the end. The resulting output is of size 1×64×801, which is represented as a 2d matrix. The details of the SINS dataset are shown in Table 1.

Classes	Samples	Samples per listener
Absence	2100	300
Calling	595	85
Cooking	763	109
Dishwashing	224	32
Eating	455	65
Other	308	44
Vacuumcleaner	98	14
Visit	385	55
Watching tv	2100	300
Working	2100	300
Total recordings	9128	1304

Table 1: Dataset distribution over all class labels for pre-processed SINS dataset [39].

8.3. Building of a custom dataset

8.3.1. Motivation

During the literature study, we found that two of the datasets that are used for the evaluation of sensor fusion methods for the task acoustic-based activity classification all focuses on domestic activity classification. Both only contain recordings in a single small indoor environment where the microphones are closely located to each other. As a result, the overlap in the sensing range of the sensors is high. Therefore, the benefit of using a WASN is limited. In real world scenarios, deploying multiple microphones in one room is often intrusive and provides limited value for activity classification. Therefore, a

novel dataset where the sensors are placed sparingly can demonstrate the efficiency of a WASN and the necessity of sensor fusion much better.

8.3.2. Procedure

To create a novel dataset with sparingly-placed acoustic sensors, we must increase the cover area, or reduce the number of acoustic sensors in the WASN. To achieve this, we built a simulation environment using the high-performance Habitat [45] 3D simulator with the SoundSpaces [46] realistic acoustic simulation extension. The top-down view of the simulation area and the placement of the acoustic sensors, and the sound source locations are visualized in Figure 2.

Figure 2: The top-down view of our new dataset, gray denotes navigable area, white denotes obstacles. Both the audio source and the listeners are located at different locations. Listener 3 overlaps with sound source 3. Compared to the previous dataset, the distance between listeners is greater, which poses a greater challenge, as the observations from different listeners can be vastly different.

Compared to the SINS dataset [39], we place the acoustic sensors across different rooms in an office building, each room only contains a maximum of one listener. This ensures that the observations of the listeners are different from each other, and it shows the efficiency of a WASN in covering a large area of interest with a limited number of sensors. Moreover, as we are working in a simulated environment, we can easily add more microphone arrays and sound sources, as well as changing the recording environments. For the experiments conducted in this report, we evenly distributed five listeners (with IDs {0,1,2,3,4}) at different locations and set five different locations from which the sound could be played. This is shown in Figure 2.

Next, the data for our simulated dataset is collected by replaying clips from microphone arrays in the SINS dataset on each sound source locations shown in Figure 2. To capture the different sounds, present in a domestic environment, we used the recordings from

microphone array 1, 4 and 8 from the SINS dataset. They are in three corners of the living room of the vacation home. The microphone array on the fourth corner was not available due to technical difficulties during the data collection [39].

We use the observations of the listeners at the five different locations as our final audio data. The size of our dataset is around 15 times larger than SINS dataset (3 microphone array recordings in the SINS dataset, replayed at 5 sound source locations). A similar pre-process with STFT and Mel-scale filterbank transformation is performed on the observed audio signals, resulting in a 2-d matrix of size 64x64 for each observation. The details of the dataset are shown in Table 2.

Classes	Samples	Samples per listener
Absence	22650	4530
Calling	13275	2655
Cooking	22650	4530
Dishwashing	6375	1275
Eating	11175	2235
Other	7800	1560
Vacuumcleaner	5100	1020
Visit	9150	1830
Watching tv	22575	4515
Working	22575	4515
Total recordings	143325	28665

Table 2: Dataset distribution over all class labels for our simulated dataset.

8.4. Results

The performance of the rule-based swarm intelligence approach is evaluated on both the SINS dataset and our simulated dataset. To test the ability of the swarm intelligence to deal with unknown sensor locations, we split the dataset into two disjoint sets based on the ID of the listeners. On our novel dataset, the first set contains observations from listeners {0,1,3}, and the second set contains observations from listeners {2,4}. On the SINS dataset, the two sets contain observations from listeners {1,4,6,8} and {2,3,7} respectively. As the rule-based swarm intelligence approach requires the sensors to

make their own decisions and then transmit it to the fusion center, a pretrained classifier is required.

During the training phase, only the first set is used. The second dataset is only used to test the performance of the model on unseen listener locations on the test phase. We denote the first set as the 'Seen' dataset and the second set as the 'Unseen' dataset. As our dataset is class imbalanced, the macro-f1 score is used to measure the performance of the baselines and our methods to account for the minority classes. As the rule-based swarm intelligence baselines are deterministic, repeating the experiment wouldn't change their results.

Category	Methods	SINS			Simulated dataset		
Sensor Cluster		(1,4,6,8) Seen	(2,3,7) Unseen	(1,2,3,4,6,7,8) Seen+Unseen	(0,1,3) Seen	(2,4) Unseen	(0,1,2,3,4) Seen+Unseen
Lower bound	Pretrained classifier	80.8%	75.4%	78.5%	88.6%	87.4%	88.1%
Decision-level fusion	Majority voting	82.0%	79.1%	81.5%	90.7%	87.5%	91.9%
	Highest confidence	**82.6%**	78.7%	81.4%	90.8%	**90.0%**	91.0%
	Sum of confidences	**82.6%**	**79.7%**	**81.8%**	**91.4%**	89.4%	**92.5%**

Table 3: Comparison of test f1-score on different set of acoustic sensors. A pretrained classifier is trained on 'Seen' set of sensors. The 'Unseen' set of sensors are never used in the training phase of the sensor fusion methods that requires training. The majority voting, the highest confidence and the sum of confidence fusion strategy are rule-based and deterministic, thus, no standard deviations are reported. Bolded text denotes the best results.

We observe that using the rule-based, decision-level fusion methods, one can already improve the macro-f1 score of the domestic event classification task. Compared to the lower bound, which is that each sensor makes their own decision, the accuracy can be improved by several percents.

The improvement is larger on the 'Unseen' dataset. This denotes that using rule-based, decision-level sensor fusion is important for real-world use cases where the sensors might be moved to unseen locations.

Finally, we see that the overall performance of the simulated dataset is higher than that of the SINS dataset. We hypothesize that this is due to the difference in dataset size, in particular the increase in the amount of data for the minority classes might have helped

the pretrained model to learn generalizable knowledge, therefore achieve a higher macro-f1 score.

9. Deep-learning-based Swarm Intelligence

9.1. Implementation of baseline

The rule-based approach only works with decision-level fusion methods, where the output format is intuitive. However, the intermediate feature vectors that the intelligent nodes in the swarm use, conveys additional information. There is no rule-based approach to fuse the intermediate feature vectors, as they typically are opaque to human experts.

To efficiently utilize the rich information from the intermediate feature vectors, a deep-learning-based sensor fusion method can be used. We implement two deep-learning-based sensor fusion baselines from prior research and explain them below.

9.1.1. Classifier stacking

The classifier stacking fusion strategy learns a random forest classifier using the confidence scores of the sensors. It can be described as:

$$y'_{final} = \mathcal{F}(\{c(x)_0|c(x)_1|\dots|c(x)_k\})$$

However, this method tends to overfit to the position of the inputs. To alleviate this, Kürby et al. [15] proposed to sort the confidence vectors by their maximum confidence. Thus, we will have an order $\mathcal{M}$ where $\mathcal{M} = \text{argsort}\ \max_y c(x)_k$. The equation is then transformed into

$$y'_{final} = \mathcal{F}(\{c(x)_{\mathcal{M}_0}|c(x)_{\mathcal{M}_1}|\dots|c(x)_{\mathcal{M}_k}\})$$

As both studied previous works related to classifier stacking [14], [15] fusion strategy did not provide parameters of the random forest meta-classifier, we conducted a hyperparameter tuning search using a grid-based strategy. The specifics of the grid we used for tuning can be found in Table 4.

Hyperparameters	Grid
n_estimators	{100,200,500,1000,1500,2000}
max_features	{"sqrt","log2",None}
max_depth	{20,40,60,80,100,None}
bootstrap	{True,False}
criterion	entropy
min_sample_split	2
min_sample_leaf	1

Table 4: Grid search specification for the classifier stacking baseline.

9.1.2. CNN-based feature-level fusion

For feature-level fusion with audio data, both Xu et al. [43] and Li et al. [16] used a Convolutional Neural Network (CNN) to transform the sensor observations to latent features. We have implemented the CNN-based feature-fusion framework from Li et al. [16]. It is chosen as their input format is similar to ours (2d matrix), while Xu et al. [43] directly uses the raw data without additional pre-processing steps, and uses multiple channels instead of mono-channel audio fragments (3d matrix).

The framework of Li et al. consists of 4 convolution layers, 4 max-pooling layers and 2 FC layers. Each convolution layer is followed by a max-pooling and ReLu activation layer. The number of filters of the convolution layers are 16, 32, 32 and 64, respectively. The kernel size of the convolution layers is 5×5 with a step size of 1. For the max-pooling layers, they are 2×2 and 2, respectively. Each convolution layer uses a zero-padding of size 2. The number of the outputs for the FC layers are 256 and 10, respectively. A dropout layer [47] with 50% probability is applied after the first FC layer to mitigate overfitting.

The sensor observations are first processed into a 2d matrix of size 64×801. The details of this process is mentioned in section “Experiment setting”. The first two convolution layers are used in parallel on all sensor data to extract the latent features. After that, the latent features from all the sensors are concatenated and forwarded to the remaining layers of the network for classification. For a fair comparison, we experimented with the

same hyperparameters used by our framework (described in section "Experiment setting"), and the hyperparameters provided by the authors, i.e., Adam optimizer with 1e − 4 learning rate and 24 batch size. The best results were reported for the experiments.

9.2. Feature-level fusion in wireless acoustic sensor networks with graph attention network for classification of domestic activities

Over the past decade, hardware advances have made processing information on-device possible (smart sensors) [48]. One of the areas of interest is the gathering and processing of audio data. Audio is highly informative for tasks such as speech recognition [40], city monitoring [49], event classification [13], [39], [50], [51], and fault diagnosis [16], [43], [44]. A camera-based sensing system may raise privacy concerns in home and office environments, and it is dependent on environmental conditions such as illumination, smoke, and occlusions. In contrast, an acoustic-based sensing system is often less intrusive, and it can be deployed in less favorable environmental conditions [52], [53]. To achieve a higher classification accuracy, sensor fusion techniques are often used to increase the spatial resolution of the information by aggregating data from multiple sensors [13]. Our work focuses on sensor fusion with wireless acoustic sensor networks (WASNs). A WASN contains sensors deployed at different locations in an environment, each sensor captures the audio signal that occurs at its location. By using multiple acoustic sensors, WASNs can cover a large area of interest. The task of sensor fusion requires the sensor to transmit its data to the fusion center. The fusion center combines the (processed) data from all the sensors to obtain a more comprehensive view of the environment and makes a final decision.

Based on the model proposed by Dasarathy et al. [22]. Sensor fusion techniques can be categorized into three different levels:

1) data/signal-level fusion,
2) feature-level fusion,
3) decision-level fusion.

They take as input:

1) raw data,

2) processed latent features,
3) decisions, from the sensors, respectively.

Figure 3 demonstrates these three categories of sensor fusion, and their interactions with the remote fusion center. Typically, the size of the raw data is much larger than the latent features, which in turn are larger than the final decisions of a sensor. Therefore, with each increase in fusion level, the amount of information that needs to be transmitted goes down drastically. The inherent wireless nature of WASNs presents many challenges [48], such as the energy consumption and bandwidth usage of the sensors, the potential loss of wireless connections to the fusion center, and the selection of the best subset of sensors for downstream tasks.

To analyze the energy consumption of the sensors, we can represent each sensor by three basic layers [13]. These layers are:

1) sensing layer
2) processing layer
3) communication layer

Previous studies [13], [27] have found that the communication layer dominates the overall energy consumption of a wireless sensor. In Dekkers et al. [6], the sensing layer uses less than 100 millijoules (mJ), the processing layers uses less than 102 mJ, while the communication layer has a variable cost. When transmitting raw audio data, the communication layer costs more than 103 mJ, when transmitting local decisions, it costs less than 100 mJ. Thus, when the transmission data size is reduced, the energy consumption is also reduced, which increases the battery life of the sensors. Therefore, a decision-level fusion approach can ensure the best battery life of the sensors.

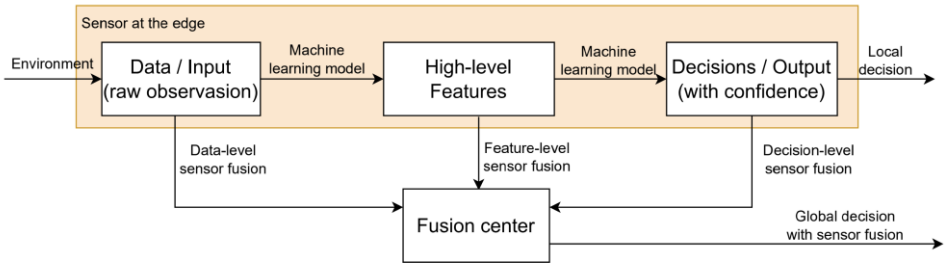

Figure 3 Three different category of sensor fusion strategies. Data-level sensor fusion requires the transmission of raw observation. Feature-level fusion requires the sensor to process the raw data, and then transmit the high-level features. Decision-level fusion requires the sensor to produce a local decision (e.g. confidence of each class in a supervised classification task.) and transmit their decision to the fusion center. The fusion sensor aggregates information from multiple sensors to provide a global decision.

Next, to solve the issue of potential unstable connections, simple heuristic-based approaches, such as max or sum, can be used to aggregate the local decisions of the sensors [13], [41]. These operations are flexible in the number of inputs, which ensures that the system can still output a prediction, even if some sensors fail to transmit their data.

Although heuristic-based decision-level fusion ensures the lowest energy consumption and the flexibility in number of input sensors, the information that is available at the fusion center is limited. Using feature-level fusion can yield better results, as the latent features contain more information compared to the local decision. However, existing methods for feature-level sensor fusion require the sensors to send more data compared to decision-level fusion, and often assume a fixed number of sensors [16], [43], [44].

Graph neural networks (GNNs) can be used to build deep learning models that can learn from graph-structured data. The spatial-based GNNs, such as graph attention networks (GATs) [54], are capable of learning from graphs that have a variable number of nodes. At each layer, a spatial-based GNN transforms the features of all nodes based on their own feature and the features of its neighbors. Based on this property, we propose our new feature-level fusion framework, visualized in Figure 4. In our framework, a fully connected graph is constructed, where each node contains the latent features

extracted by a sensor. As we use a GAT to aggregate the latent features, our framework maintains the flexibility of dealing with a variable number of sensors. Moreover, the attention mechanism of the GAT alleviates the challenge of selecting the optimal subset for sensor fusion [48] by assigning a higher importance to the nodes with more useful information.

Next, we introduce the message condensation layer at the edge. It is a small, fully connected (FC) layer that extracts the most useful features for sensor fusion from the existing latent features. By controlling the width of the message condensation layer, our framework can maintain a low communication cost and bandwidth usage. Therefore, compared to the decision-level fusion methods, our framework solves the core challenges of sensor fusion with a WASN, while using the more informative latent features from the sensors. We summarize the contributions of our work as follows:

- We propose a new framework for feature-level fusion using a GNN at the fusion center. The framework is visualized in Figure 4 and explained in section Methodology. Our framework maintains the flexibility and low communication cost of decision-level fusion but utilizes the more informative latent features instead of local decisions.

- We evaluate our framework on the SINS [39] dataset. Empirical results show that our framework outperforms multiple decision-level fusion strategies. The results confirm our hypothesis that the latent feature conveys more information compared to local decisions. Moreover, our framework outperforms convolutional neural network (CNN)-based [16] and fully-connected (FC) layer-based feature-level fusion strategies with notably reduced communication cost.

- We evaluate the performance of the framework for different messages sizes. This size is measured by the total floating-point number (FP) transmitted from the sensor to the fusion center, it corresponds directly to the communication cost of the sensors [6]. We observe that our framework outperforms the decision-level baselines at different message sizes.

- We demonstrate the generalizability of our framework by testing it on unseen sensor locations. Empirical results demonstrate a notable improvement in accuracy over the decision-level fusion strategies. Our framework is the first feature-level fusion framework for the task of acoustic-based event classification with a WASN that can handle a dynamic number of sensor inputs.

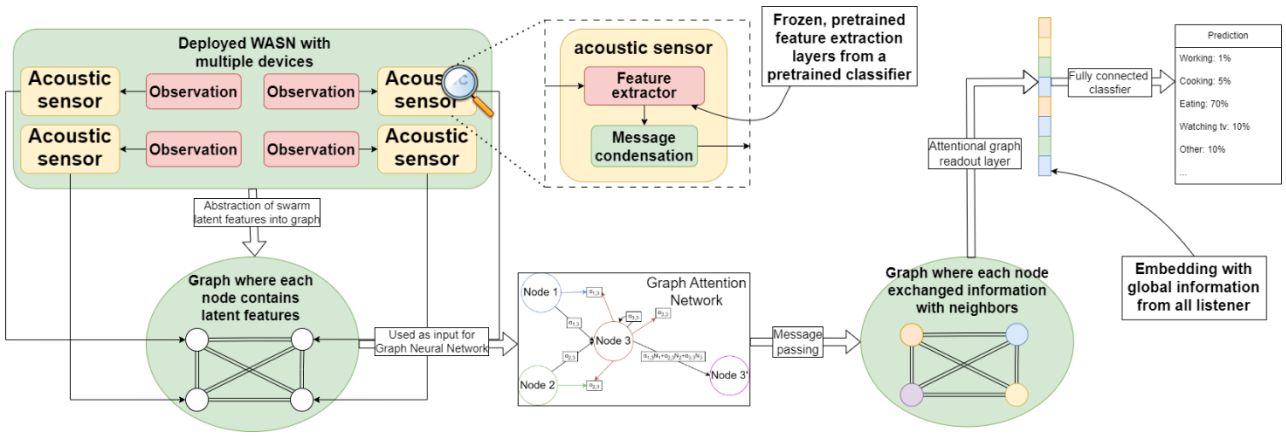

Figure 4: The architecture of our new framework. A pretrained, frozen classifier trained on single sensor observations is used as feature extractor. We remove the last layer of the classifier and replace it with a message condensation layer that condenses the latent feature vector to the desired size. The condensed feature is sent to the remote fusion center, which constructs a graph based on the sensor's connectivity. This graph is used as input to a Graph Attention Network [18]. Finally, the graph is transformed back into a latent feature vector using the attentional graph readout layer [19]. The final feature vector contains the global information of all sensors. This vector is passed through a fully connected layer for classification.

9.2.1. Methodology

Problem description

We tackle the problem of classifying domestic activities with wireless acoustic sensor networks (WASN) using feature-level fusion and deep learning models. We assume a deployed WASN with k acoustic sensors at different locations. Next, we denote the observations recorded by the sensors as x, and the function that maps the observation to the processed latent features as $f(x): R^{|x|} \rightarrow R^{d}$, such that $f(x_k)$ denotes the latent features extracted from signals captured by the sensor k. Finally, the domestic event that corresponds to the observations is denotes as $y \in Y$.

Next, we define the feature-level fusion function $\varphi : R^{k\times d} \rightarrow Y$ that maps the latent features of the k sensors with size d to the label space Y. The feature-level fusion function φ is able to accept a varying number of total sensors k for additional sensor deployments and robust against sensor failures. Finally, we denote the final decision of the feature-level fusion function $\varphi(\{f\ (x_1),\ldots,f\ (x_k)\})$ as y'. The objective of feature-level fusion is to find the optimal function φ^* such that:

$$\phi^* = \underset{\phi}{argmin}\ \ L(y', y)$$

Here, L denotes the objective function we use to compare the final decision and the ground-truth label. In this work, cross-entropy is used as an objective function.

From WASN to graph

To address the problem of domestic activity classification with WASN, we represent the WASN with all its sensors as graph-structured data. specifically, the WASN itself can be seen as an undirected graph, where each acoustic sensor corresponds to a node vi and the edges can be constructed using spatial proximity, i.e., there is an edge between two sensors when their distance is below the threshold D. In this work, we consider a fully connected graph, which corresponds to an infinitely large threshold D. The observed acoustic signals, the latent features, or the local decisions from the sensors i can all be used as the initial node embedding h_i.

Graph attention network

A graph attention network (GAT) [54] is a graph neural network (GNN). GNNs are neural networks designed to deal with graph-structured data. In particular, GAT transforms the input graph using several graph attention layers (GALs). Each GAL is associated with a learnable weight matrix $W \in R^{F'\times F}$, and a learnable attention mechanism a: $R^{F'} \times R^{F'} \rightarrow R$. GAL takes as input the edges E of the graph G and the set of node features $h = \{h_0, h_1, \ldots h_N\}, h_i \in R^F$. It outputs a new set of node features $h' = \{h'_0, h'_1, \ldots h'_N\}, h'_i \in R^{F'}$.

To obtain h', GAL first applies a linear transformation using W on all the input node features. Next, the attention mechanism a is used to compute the attention coefficient α_{ij} for each edge $e_{ij} \in E$. The process is described as:

$$\alpha_{ij} = a(\mathbf{W}h_i, \mathbf{W}h_j)$$

After obtaining the attention coefficient α for each edge, GAL computes for each node v_i, the softmax value of α_{ij} for each edge $e_{ij} \in N_i$, where N_i denotes the neighborhood, or the first-order neighbors of the node i. We denote the softmax values for the neighborhood attention coefficient α_{ij} as α'_{ij} and defines the process as:

$$\alpha'_{ij} = softmax(\alpha_{ij}) = \frac{exp(\alpha_{ij})}{\sum_{k \in \mathcal{N}_i} exp(\alpha_{ik})}$$

Finally, the new node feature h'_i is obtained by a weighted sum of the node features from the neighborhood Ni with α'_{ij} as weights, followed by a non-linear activation function σ. I.e., $h'_i = \sigma(\sum_{j \in N_i} \alpha'_{ij} W h_j)$. Figure 5 visualizes a simplified version of the aforementioned process without the linear transformation, the softmax, and the non-linear activation stages. A multi-head attention strategy was proposed in the original paper [54] to stabilize the learning process. It involves initializing multiple sub-GALs in a layer to transform the same input, and then concatenating their outputs as the final outputs of the layer.

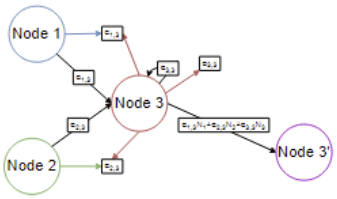

Figure 5: A simplified version of the graph attention layer, the figure illustrates the computation of the node features for node 3 (N3), where node 1 (N1) and node 2 (N2) are the first-order neighbors. The attention coefficients α are computed for each neighbor and node 3 itself. It is then used as the weight to compute the weighted sum of node features as the new node feature for node 3.

As a GAL transforms the features for each node separately, adding or removing a node in the graph does not change the process of GAL. This property makes GAT ideal for

domestic activity classification with WASN, where the connections can be unstable and interrupted.

9.2.2. Framework

We propose a novel feature-level fusion framework for classification of domestic activities with a WASN using a GAT. It is visualized in Figure 4. In this section, we explain different parts of the framework in detail.

On-device feature extractor

Our framework starts with a deployed WASN. The audio sensors are deployed in the environment and equipped with a pretrained, frozen classifier that is used as the feature extractor. To imitate the constraint in computation resources at the edge devices, we used a small convolutional neural network (CNN) with two convolution layers and one fully connected (FC) layer, as visualized in Figure 6. We followed the architecture of Dekkers et al. [13] and used 1D-convolution layers, followed with max-pooling and ReLu activation. A global average pooling layer is used at the end to aggregate the remaining dimensions of each row.

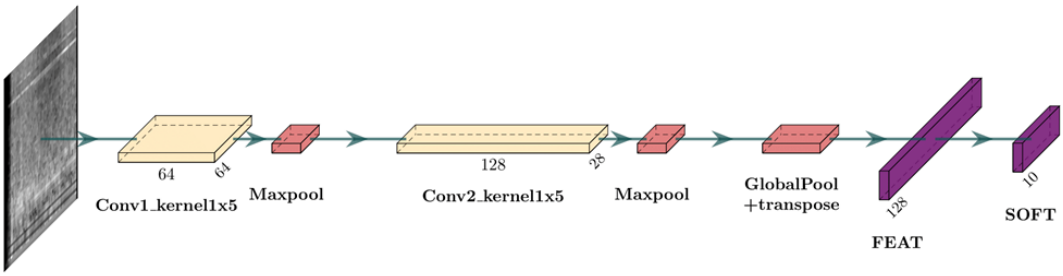

Figure 6: Architecture of the pretrained classifier. The input is the audio signal, transformed to mel-spectrogram, and represented as a 2d matrix of size 64×801. The 1D-convolution layers have kernel size of 5, the maxpool layers have kernel size of 5 and stride of 2. The first convolution layer expands the input to 64 channels. The second convolution layer expands the input to 128 channels. A global average pool layer is used to aggregate the remaining dimensions after the second convolution layer to a feature vector. The feature vector is used by an FC layer, which predicts the probability of the 10 classes. A ReLu activation and batch-norm layer is used after each maxpool layer.

This classifier is pretrained on single sensor observations, transformed to a Mel-spectrogram and represented as a 2d matrix, we explain our pre-processing step in

section IV-A. During the pretraining of the classifier, the final classification layer is used to predict the 10 domestic activities to compute the classification loss. We train this classifier on the observations of all listeners in the dataset, 60% of the data is used for training, 20% for validation and 20% for testing, the classifier is trained for 100 epochs on the training set with batch size of 64 and an AdamW optimizer [55] with $1e-3$ learning rate and 0.01 weight decay. A dropout layer [47] with 50% probability is used before the classification layer as an additional regularization. We pick the epoch with the best validation loss and use it as the final model for feature extraction on the sensors.

Message condensation

After the pretraining finishes, we remove the final classification layer and replace it with a randomly initialized fully connected layer that maps the flattened features to a lower dimension. This is the message condensation layer in Figure 4. The output of the message condensation layer will be the information that the edge device must transmit to the remote fusion center. Thus, by adding the message condensation layer and setting its output size, we introduce a mean to control the communication cost for the edge devices. This layer is not frozen, and it is trained as a part of our framework, along with the graph neural network that performs the sensor fusion and is shared by all the deployed sensors.

A PReLu activation layer [56] is used after the message condensation layer to prevent the dying neuron problem of the ReLu activation layer [57]. It is the issue where the neuron outputs zero values for all the input data and the backpropagation can't effectively update the neuron anymore. In our case, when the size of the condensed messages is small, the dying neuron will have a great impact on the performance of our model.

Finally, we note that any computation after the message condensation layer occurs at the remote fusion center. Thus, additional computations after this point do not affect the battery life of the edge devices anymore.

GNN-based sensor fusion

At the remote fusion center, a fully connected graph is constructed. Each node in the graph represents a sensor in the WASN, and the node features are the condensed messages. A two-layer graph attention network [54] is used to transform latent features of each node, which only contain local information, into latent features that incorporate information from neighboring nodes. The first layer outputs 16 hidden features, and the second layer outputs 32 hidden features. Both layers use 4-headed attention to stabilize the learning process, i.e., each attention head outputs $1/4$ of the hidden features of the layer. A PReLu activation [56] is used after each graph attention layer.

We have chosen a shallow architecture to prevent the oversmoothing phenomenon: Li et al. [58] has shown that if a GNN is deep with many graph convolutional layers, the output feature may be oversmoothed, as all nodes will observe a similar feature, and the node features will become indistinguishable.

We also introduce the node drop-out mechanism in this step. I.e., there is an $x\%$ chance for each node to be removed from the constructed graph during the training phase. This mechanism will never remove all nodes. It encourages the model to be robust with any number of nodes, which improves the model's ability to generalize to future node deployments and sensor failures. The detailed list of the hyperparameters used for the experiments is provided in section Experiment setting.

Graph-level prediction

After the feature extraction with a graph attention network, an attentional graph readout layer [59] is used to extract a final embedding with global information from all nodes. It computes an attention score for each node in the graph and transforms all the features one more time with a fully connected layer. Then, after applying a softmax on the attention scores, a weighted sum of the transformed node features is computed as the final embedding for the graph. The attentional readout is used as it performs empirically better than the simple sum or mean readout [59]. To further facilitate the generalization of the model, a feature dropout layer [47] is used on the final embedding. This dropout is different from the node dropout, as the previous one enhances the robustness against

various number of nodes, while the feature dropout aims to reduce the probability of the model overfitting at the training data samples. After that, the final embedding is forwarded to a last FC layer for the classification of the event observed by the WASN.

9.2.3. Additional Baselines

Apart from the implemented decision-level and feature-level fusion baselines, we construct a concatenation-based feature-level fusion framework with the same frozen pre-trained classifier as our framework. This baseline does not include the message condensation layer and transfers the full latent feature back to the fusion center. Next, the latent features of all the sensors are concatenated together and forwarded to three FC layers for classification.

We test two different widths of the FC layer size, the width of the FC layers on the first variant is set to {16, 32, 10}, respectively. This variant has a similar number of parameters compared to our framework. The second variant has FC layers with width {256, 128, 10}, respectively. This variant is chosen such that each FC layer reduces the latent representation size by $1/2$. Both variant use a PReLu activation and batch normalization after the first two FC layers, and a dropout of 50% before the last FC layer. We trained this network on the same hyperparameters as our framework. They are denoted as FC-small and FC-large in the results.

9.2.4. Experiments setting

We evaluated our framework, together with the baselines, on the task of classifying domestic activities with sensor fusion on the SINS dataset [39]. To demonstrate the generalization of our framework to unseen listener locations, we split the dataset into two disjoint sets based on the ID of the listeners. The two sets contain observations from listeners {1,4,6,8} and {2,3,7} respectively. During the training phase, only the first set is used. The second set is only used to test the performance of the model on unseen listener locations on the test phase. We denote the first set as the 'Seen' dataset and the second set as the 'Unseen' dataset. As our dataset is class imbalanced, the macro-f1 score is used to measure the performance of the baselines and our methods to account for the minority classes.

We split the 'Seen' dataset into three parts for the training and evaluation purposes. 60% of the 'Seen' dataset is used for training, 20% for validation, and 20% for testing of the baselines and our framework. The data are divided based on the timestamps, i.e., the observations of all listeners at timestamp x can only be in one of the three sets. This prevents similar observations from existing across the training, the validation, and the test set. The similar timestamp split is performed on the 'Unseen' dataset as well, however, we will only be using the 20% test set of the 'Unseen' dataset to test the generalization of the experimented methods.

During the training phase, an AdamW optimizer [55] with $1e-3$ learning rate and 0.01 weight decay were used. The batch size is set to 16. The node dropout and feature dropout rate has been set to 50%. The models are trained for 100 epochs with cross-entropy as a loss function. The epoch with the lowest validation loss is used for the evaluation of the test set performance. Finally, we used a weighted random sampler with sample weights corresponding to the inverse of the class size to alleviate the class imbalance issue of the SINS dataset.

For the classifier stacking method, we follow the same protocol as Kürby et al. [15] and use $2/3$ of the training set for the base classifier, $1/3$ for the training of the random forest meta-classifier.

For our message condensation layer, we experimented with message size of 4, 6, 8, 10, 20 and 40. We have 10 classes in our dataset, thus, a message of size 10 corresponds with a communication cost of sending 10 FPs, which is equal to the message size of sum of confidence and classifier stacking decision-fusion method.

All the experiments are repeated 5 times, the mean and standard deviations of the reported results are visualized for the learning-based methods, i.e., classifier stacking and our GNN framework. Finally, for the classifier stacking and CNN/FC-based feature fusion, as they can't perform inference with different input size, only the results of 'Seen' set of sensors are reported.

9.2.5. Results

We report the main result of our experiment from two different aspects. First, we compare the performance of the framework with the condensed message size of 6 FPs against the decision-level fusion baselines to demonstrate the performance of our framework at a similar communication cost with the baselines. Then, we compare our framework at different message size constraints against the best performing decision-level fusion baseline to demonstrate the performance of our framework at varying communication costs. We report the results on the 'Seen', 'Unseen' and the joint 'Seen+Unseen' test sets. Additionally, the performance of the pretrained classifier is reported as a lower bound for the sensor fusion methods. The focus of the table is on how much the sensor fusion methods improves the results compared to the pretrained classifier. This pretrained classifier is used to determine the local decision for decision-level fusion methods, and latent features for our framework and the 'FC-{small, large}' baselines. The 'Li et al. [16]' baseline uses a 2D-CNN as feature extractor that is trained during the sensor fusion process. Next to the main experimental results, we also conducted additional ablation study and scalability experiments.

Matching communication cost

We first demonstrate the performance of our framework at a comparable communication cost to the decision-level fusion baselines. This means that the message condensation layer will output a condensed feature of size 6. This is smaller than the message size required by the sum of confidence and the classifier stacking baselines. The macro-f1 score of the three test sets are reported in Table 5.

We note that the baseline from Li et al. [16] has a higher message size. The output of the second parallel convolution layer of Li et al. [16] has 88704 hidden units. Thus, the communication cost of Li et al. [16] is significantly higher compared to the other experimented methods. Next, while the majority voting and highest confidence baselines have a lower communication cost (less than 1 FP and 1 FP, respectively), they also delivered a worse performance compared to the sum of confidence baseline.

Referring back to the communication cost analysis done by Dekker et al. [13], we observe that the communication cost at the message size of 6 FP is already lower than 10^0 mJ, which is two magnitude lower than the energy consumption of the processing layer [13]. For the feature-level fusion method proposed by Li et al. [16], the message size of 88704 single precision FPs corresponds to the transmission of at least $3e5$ bytes. The energy consumption at that message size is between 10^1 and 10^2 mJ, similar to the energy consumption of the processing layers [13]. This means that their communication cost contributes notably to the energy consumption of the sensor, while the communication cost of sending a message with size less than 10 only contributes less than 1% of the total energy consumption of the sensor.

Comparing the test f1 score of our framework with the baselines, we see that, at a similar communication cost, our GAT-based feature fusion framework outperforms the decision-level fusion baselines in all settings. Specifically, our framework outperforms the best decision-level fusion baseline by at least 1.5% (absolute) in macro-f1 score. In the most realistic setting 'Seen + Unseen, our framework outperforms the second-best baseline by 1.9%.

Category	Methods	SINS		
Sensor Cluster		(1,4,6,8) Seen	(2,3,7) Unseen	(1,2,3,4,6,7,8) Seen+Unseen
Lower bound	Pretrained classifier	80.8%	75.4%	78.5%
Decision-level fusion	Majority voting	82.0%	79.1%	81.5%
	Highest confidence	82.6%	78.7%	81.4%
	Sum of confidences	82.6%	79.7%	81.8%
	Classifier stacking [41]	76.7% ± 0.3%	-	-
Feature-level fusion	Li et al. [11] (message size 88704 FPs)	75.2% ± 4.8%	-	-
	FC-small (message size 128 FPs)	82.2% ± 1.1%	-	-
	FC-large (message size 128 FPs)	82.6% ± 1.9%	-	-
Ours	Ours (message size 6 FPs)	**84.5% ± 0.4%**	**81.2% ± 1.4%**	**83.7% ± 0.7%**

Table 5: Comparison of test f1-score on different set of acoustic sensors. A pretrained classifier is trained on 'seen' set of sensors. The unit of the message size is the storage space required for a floating point number. The 'unseen' set of sensors are never used in the training phase of the sensor fusion methods that requires training. The majority voting, the highest confidence and the sum of confidence fusion

strategy are heuristic based and deterministic, thus, they do not have a standard deviation. Bolded text denotes the best results and underlined text denotes the second best results.

We also see that the classifier stacking baseline performed the worst of all the decision-level fusion baseline. Following the protocol used by Kürby et al. [15], we only use 2/3 of the training data for the pretraining of the classifier. We hypothesize that this splitting of training data for the training of the classifier and meta-classifier harms their performance severely. Moreover, as it requires a fixed sized input, it can not handle the other two sensor clusters.

Next, for the feature-level fusion baselines, we observe that the CNN-based feature-level sensor fusion framework from Li et al. [16] performed worse than the lower bound. This denotes that a 2D-CNN based approach is not suitable for dealing with mel-spectrogram images created by audio signals. Both FC-based feature-level fusion baselines achieved a similar result compared to decision-level fusion baselines. However, compared to our framework, which has additional regularization from the node dropout layer, they still performed worse. Moreover, they share the same issue with the classifier stacking baseline, all of them can only deal with a fixed number of sensor inputs.

Varying communication cost

Our framework introduces the message condensation layer to constrain the energy consumption and bandwidth usage of the communication layer of the wireless acoustic sensors. A direct concern to this layer is that the performance of the model might be affected by the message condensation layer. To address this concern, we conducted the same experiment with our framework with a different width for the message condensation layer. The results are compared against the second best performing sum of confidence fusion method and visualized in Figure 7.

In Figure 7, we demonstrate the performance of our framework at different message size on the SINS dataset. We use the sum of confidence as a baseline for comparison, as it is the best performing decision-level fusion method. We observe that, both an increase and a decrease in the message size harms the performance of the model on the ‘Seen’ and ‘Unseen’ test set. Our hypothesis is that, a lower message size is

insufficient to convey all the useful information to the fusion center, while a higher message size leads to overfitting. I.e., the constraint in the message size acts as a form of regularization in our framework. Despite the decrease in performance, all models consistently outperforms the baseline in the most realistic 'Seen + Unseen' test set. Finally, the performances of the models on the 'Seen + Unseen' test set mimics their performance on the 'Seen' dataset, which is available at the training time. I.e., we can fine-tune the message size before the deployment of the model and expect a similar increase after deployment compared to the decision-level fusion baselines.

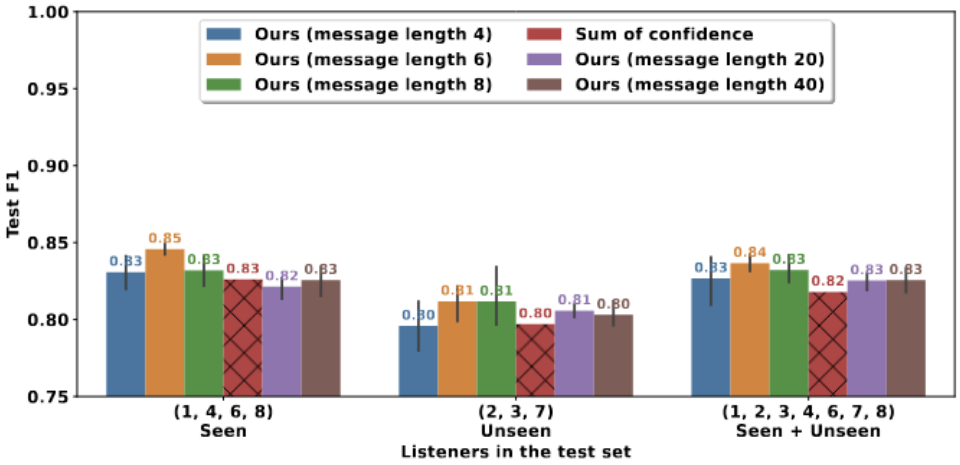

Figure 7: The macro-f1 score of our framework with different message sizes, compared to the sum of confidence baseline, denoted by the hatched red bar. Our method outperforms the baseline with a lower communication cost (message size 6 FPs). Reducing or increasing the message size cause a decrease in performance. However, all our models consistently outperform the decision-level baseline in the most realistic 'Seen + Unseen' setting.

Next, we notice that our model still performs similar to the sum of confidence baseline at a message size of 4. The message size of the sum of confidence baseline is 10. Our framework is the first feature-level fusion framework that achieves a communication cost that is similar to the decision-level fusion methods.

9.2.6. Ablation study

To study the efficiency of each part of our framework, we conducted additional ablation studies. We performed 4 additional experiments on our framework, each experiment removes one component in our framework to demonstrate their contribution to the final performance. These experiments remove:

1) the usage of a frozen, pretrained classifier as feature extractor,
2) the message condensation layer,
3) the attention based aggregation,
4) the node dropout regularization, respectively.

The results are reported in Table 6.

Methods	SINS		
Sensor Cluster	(1,4,6,8) Seen	(2,3,7) Unseen	(1,2,3,4,6,7,8) Seen+Unseen
Ours (message size 6)	**84.5% ± 0.4%**	**81.2% ± 1.4%**	**83.7% ± 0.7%**
- frozen pretrained extractor	82.6% ± 0.9%	79.2% ± 1.4%	81.9% ± 1.6%
- message condensation	83.8% ± 0.8%	81.1% ± 1.4%	83.1% ± 1.2%
- attentional aggregation	83.1% ± 2.2%	80.1% ± 1.8%	82.4% ± 1.7%
- node dropout	82.5% ± 2.4%	79.6% ± 2.7%	82.7% ± 2.0%

Table 6: Ablation study of our framework, the framework with a message size of 6 is used as the baseline. Each row below the baseline denotes an experiment where the corresponding module is removed from the framework. The experiment without message condensation has a message size of 128. A global mean aggregation is used for the experiment without attentional aggregation.

From the table, we observe that each of the experimented module is crucial in achieving the reported performance. Noticeably, the usage of a pretrained, frozen classifier as a feature extractor, and the node dropout contributes the most towards the macro-f1 score of the model. The attentional aggregation and the node dropout layer both reduce the standard deviation on the 'Seen' dataset. The node dropout layer is crucial at reducing the standard deviation on the 'Unseen' and 'Seen + Unseen' datasets. Finally, the message condensation layer also contributes slightly to both the performance (macro-f1) and the stability (standard deviation) of the framework. We hypothesize that the restriction of message size can act as a form of regularization, which helps our framework to learn generalizable knowledge. When this layer is removed, the framework is more prone to overfitting the training data.

9.2.7. Conclusion

In this section, we focused on using sensor fusion for domestic activity classification using a wireless acoustic sensor network (WASN).

We analyzed the existing challenges for sensor fusion with wireless acoustic sensor networks (WASNs) in the application of domestic activity classification. The challenges include the energy consumption of the communication layers of the wireless acoustic sensors, the unstable connection from the remote sensors with the fusion center (i.e. variable number of sensors for different fusion attempts), and the selection of the best subset of sensors for sensor fusion [48].

To solve these challenges, we proposed a graph neural network (GNN) based feature-level sensor fusion framework. A graph attention network (GAT) [54] was used such that the attention mechanism can be used to learn which sensors contain important information. I.e., the best subset of sensors can be derived inherently.

In contrast with the decision-level fusion methods, our framework uses the latent features extracted from sensor observation, which contains richer information. In contrast with other feature-level fusion methods, our GAT-based framework does not rely on the concatenation operation for sensor fusion. Rather, we represent each sensor as a node in a graph. As a GNN is designed to handle graphs with a varying number of nodes, our framework can handle a variable number of sensors as well.

Next, we implemented the node dropout layer and the message condensation layer. The former encourages the GAT model to be robust for any number of input sensors. The latter condenses the latent feature extracted from the sensor observation to a desired length, reduces the communication energy consumption and the bandwidth usage of the sensors, which will prolong the sensors' battery life.

Extensive empirical experimentation on the SINS [39] dataset shows that our framework outperforms different decision-level and feature-level fusion baselines [16]. Moreover, our framework maintains the flexibility of accepting a variable number of sensor inputs. Our framework is the first feature-level fusion framework that outperforms decision-level fusion baselines while maintaining a low communication cost and the robustness to sensor outages and additional sensor deployments. The additional ablation studies and scalability experiments demonstrate the versatility of our framework for real-world applications.

9.3. Graph neural network for underwater sound source localization

In the previous section, we described how we built a GNN-based sensor fusion framework for the task of event classification. It is interesting to extend this work to the task of sound source localization (SSL), such that we can use it to further improve the performance of e.g., the ocean noise pollution monitoring proof of concept in the OpenSwarm project.

In preparation of this future work, we conducted a literature study on the related works involving sound source localization. Next, we discuss the existing real-world datasets that contains underwater sounds. Finally, we present our preliminary experiments conducted with the objective of reproducing the state-of-the-art results in SSL and provide concrete insights to achieve accurate SSL for the proof of concept.

9.3.1. Related work

A WASN can utilize different features to identify the location of the source signal [60]. Previous works have used the time-of-arrival (TOA) [61], time-difference-of-arrival (TDOA) [62], [63], [64], estimated direction-of-arrival (DOA) [65], and the steered response power (SRP) [66].

We are interested in expanding our GNN-based sensor fusion network for the task of SSL. The work of Grinstein et al. [64] is the most similar to ours. They tackle the problem of SSL in a single room by discretizing the area into a grid of size 25×25. Each grid cell is assigned a number between 0 and 1 based on the proximity with the sound source. They formulated the problem as a regression task, where the model must output values for each cell, and the objective is to minimize the discrepancy between the output value and the assigned value of each cell.

To do this, they used the spatial likelihood function [63] with the signals extracted from each pairs of microphones to obtain a set of probabilities for each cell to contain the sound source. Then, a relational network [67] is used to transform the set of probabilities into latent features. In the end, a summation operation is performed to aggregate the latent features into a final feature vector, which is used to output the final values assigned to each cell.

The TDOA between the two sensors is required to obtain the spatial likelihood function, and the estimation of TDOA is typically done by computing the generalized cross correlation [68] between a pair of signals. This means that the method developed by Grinstein et al. [64] requires the raw data of the microphones to be transmitted to the fusion center. The objective of this work is to research the possibility of utilizing a graph attention network [54] with local feature extraction to reduce the amount of data transfer between the wireless acoustic sensors and the remote fusion center, such that the energy consumption of the sensors can be reduced without a decrease in performance.

9.3.2. Dataset

In this section, we discuss our literature study on the potential datasets for the task of boat classification and localization using underwater acoustic sensors.

Dataset of Louise et al. [69]

In the proposal document of the OpenSwarm project, the convolutional neural network-based boat detection method developed by Louise et al. [69] is used as an example of a possible solution to boat classification. In their work, the dataset was collected by placing hydrophones at five different sites in the Hauraki Gulf Marine Park, New Zealand. In total, 30264 files, or 12% of the dataset were labelled manually. The author used the spectral signatures of the boat sounds to identify the presence of the boats in the spectrogram of the collected sound clips.

The dataset collected from this work overlaps highly with the objective of the ocean noise pollution proof of concept. However, after contacting the corresponding author, we were not able to obtain access to this dataset. Instead, the dataset of a follow-up work [70], which is publicly available, was provided as an alternative solution. However, the data of this work only contains unlabeled raw acoustic signals, the author suggested that we should manually label them ourselves. We believe that a better alternative must exist which contains labelled boat sounds for underwater acoustic classification.

Shipsear dataset

After failing to obtain the dataset utilized in Louise et al. [69], we conducted a study on existing underwater acoustic classification datasets. We found two datasets which can serve for our objective, namely the Shipsear dataset [71] and the Deepship dataset [72].

The Shipsear dataset [71] contains 90 recordings of 11 boat types, including fishing boat, trawler, mussel boat, pilot ship and other type of boats. The sounds were collected during autumn 2012 and summer 2013, using hydrophones near the port of Vigo, Ria de Vigo, Spain. Each recording has a length varying between 15s to 10min.

To validate the usability of the dataset, the author grouped the 11 boat types into 4 different super-classes and added the background noise collected at the Intecmar meteorological station outside of Ria de Vigo as an additional class. The acoustic signals were transformed with cepstral coefficients and a statistical classification was done using gaussian mixture models. The classification accuracy was not optimal, with the minority class only containing 7 recordings, and only achieving an accuracy of 55.5%. Due to the small dataset size, we believe that this dataset will not be sufficient to train a generalizable neural network. Moreover, following the license provided by the author, this dataset can only be used for educational purposes, which might be conflicting with the project objectives.

Deepship dataset

Compared to Shipsear [71], Deepship [72] is a more recent dataset with a focus on boat classification using deep learning algorithms. This dataset contains 265 recordings of four different boat types: cargo ship, tug, passenger ship and tanker. The length of the recordings varies from 6 seconds to 1530 seconds.

The recordings in Deepship [72] are taken between 02 May 2016 and 04 Oct 2018. The data recording site is near one of the busiest ports in Vancouver, Canada. The data was collected in 3 periods, the location and the depth of the hydrophone differs in each period.

The dataset was labelled with the help of the automatic identification system (AIS) data. The AIS can be used to locate the location of the vessel at any given timestamp, given that an AIS radio transceiver is located on the vessel. Thus, using this data, the authors of the Deepship [72] could determine when a boat is passing by the hydrophone, and give it the corresponding label based on the boat types.

To demonstrate the usability of the dataset, various experiments were conducted in their work. This includes different acoustic feature extraction steps, paired with different deep neural network architectures. The best combination reported by the authors has an accuracy of 77.5%. However, a potential disadvantage of this approach is that boats that are not being monitored by the AIS can introduce noise in the data. Thus, we hypothesize that some manual cleaning process still needs to happen to ensure that the recordings do not contain multiple boat sounds that are not in the same class.

Compared to the Shipsear [71] dataset, which has a total recording length of ~10000 seconds, Deepship [72] has a total recording length of 169440 seconds, which is ~17 times larger. The large data size makes Deepship more suitable for deep learning applications. Moreover, the dataset is publicly available online, the ease of access makes Deepship the preferred dataset for the application of underwater acoustic classification.

Passive acoustic monitoring datasets

Although Deepship can be used for the task of underwater acoustic classification, it only contains the recordings from a single hydrophone at any timestamp. Thus, it is difficult to perform sound source localization and also difficult to apply the sensor fusion framework which we developed. In an attempt to find the suitable dataset, we looked into the passive acoustic monitoring datasets, such as SanctSound [73].

These datasets contain extensive labels for the audio recordings, the purpose of the dataset is to monitor the underwater mammal activities. These datasets contain the continuous recordings from the hydrophones located at different locations, and the interval at which a certain sound is labelled. After examining the recordings manually, we found that the interval that was provided by this dataset mostly only contains the

labelled sound for a few second, followed by a long period of inaudible sounds. Thus, the dataset requires a lot of cleaning before it can be used for the training of deep learning models.

Simulated datasets

Our literature study concludes that there is currently no real-world dataset that is suitable for the task of underwater SSL. The difficulty in collecting usable real-world data for SSL is also mentioned in Jin et al. [74].

The alternative will be to generate a simulated dataset. This is done in numerous previous works in underwater SSL [74], [75], [76]. We believe that a simulated dataset will be the most suitable to collect the required amount of data for the training of the deep learning models. Thus, further examination into simulation such as the Kraken normal mode program is required.

9.3.3. Preliminary results

Quantitative results

As an initial baseline, we recreated the method proposed by Grinstein et al. [64]. As the dataset is not provided by the author, we used a list of 128 wav files of length 1 second collected from the internet as the source signal. Some examples of the sounds in the list include alarm, birds, bell, engine, person, fan and horn.

A simulated training set of size 30000 is generated using the corresponding simulator code provided by the author. This dataset contains recordings of 5 microphones located at a rectangular room with randomized length and width, the microphones and the sound sources are placed at a random location. Figure 8 shows an example of the data in the simulated dataset, the figure shows the room size, sound source, and the microphone placements. A validation and test dataset of size 1000 is generated with the same configuration.

To recreate the result reported by the authors, we trained a model with the pairwise spatial likelihood function [63] as input, and used the mean squared error as the regression loss for the output value of each cell. However, as the room is discretized

into a grid of size 25×25, we believe that it is more logical to frame it as a classification problem where the model must classify which cell contains the sound source.

To implement the model as a classification task, we located the cell where the sound sources are and used the ID of the cell as the class label. We have in total 625 classes, each one corresponding to one of the 25×25 cells. A cross-entropy loss is used as the objective function that the model is trying to minimize.

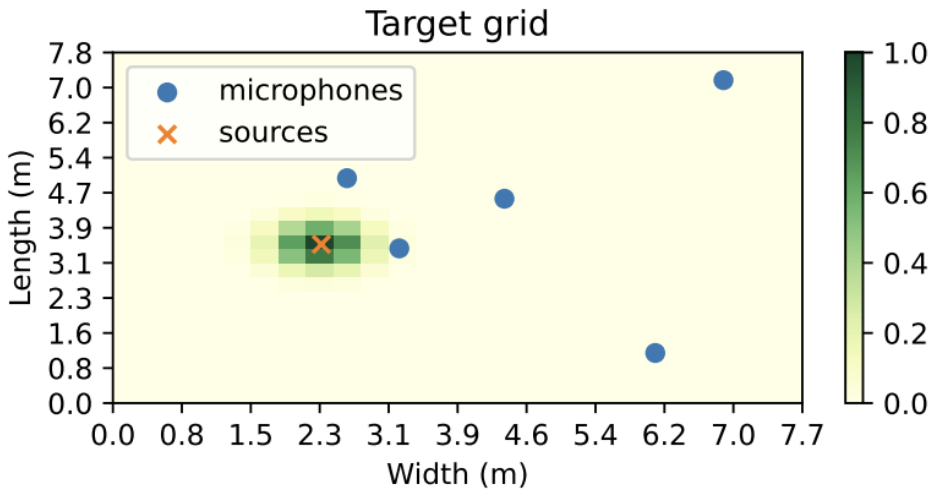

Figure 8: An example of the room configuration in the simulated dataset from the simulator provided by Grinstein et al. [64].

Both models are trained for 100 epochs with a batch size of 128 and an Adam optimizer with learning rate 0.001, the learning rate decays at epoch 1, 30 and 60 by 1/2. An early stopping strategy is applied with a patience of 30 epochs. The results are reported in Table 7. We use the average distance from the estimated coordinate (the top-left of the cell with the highest output value) to the sound source as a metric of the model's performance.

	Simulated dataset
Grinstein et al. with MSE loss	0.220
Grinstein et al. with CE loss	0.325

Table 7: Average distance from the cell with the highest output value to the ground-truth source coordinate in meter, the simulated dataset has a training size of 30000, validation size of 1000 and test size of 1000.

From the preliminary experiment, we see that the model with classification task underperforms the model with a regression task. We believe that this is partially due to the high class count and the relative small dataset that has been generated for this experiment. As an example, the Tiny-Imagenet dataset [77] has 200 classes and 500 training images per class. While our simulated dataset currently has 625 classes and on average 48 training images per class. Thus, a regression based on the distance to the ground-truth sound source coordinate may provide more training information and achieve a better performance with less training samples.

Qualitative results

To assess the qualitative result of the grid-based SSL solution, we extended the simulator by Grinstein et al. [64] with the ability of generating a timeseries of data where the sound source moves over time.

A total of 1000 data points was generated, and the SSL model trained with mean squared error as loss was applied to obtain the estimated sound source location at each timestamp. Both the target and the estimation are visualized in Figure 9.

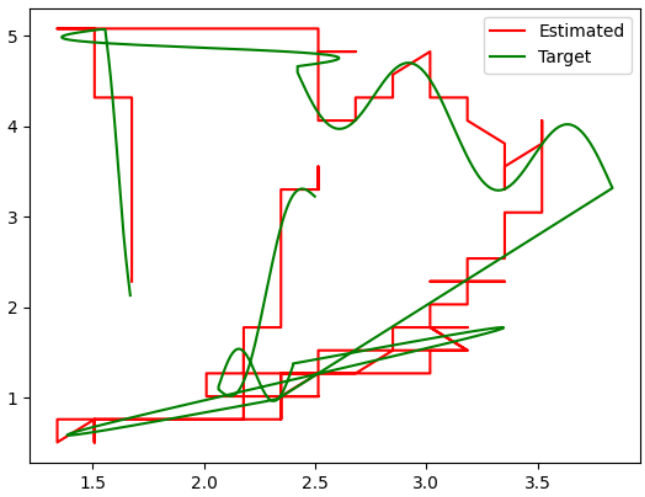

Figure 9: The estimated and the target coordinate on the simulated dataset with moving sound source. The estimation was made by the SSL model trained with mean squared error as the loss function. The x and y axis denote the length and width of the room in meter.

From Figure 9, it is clear that a grid-based SSL solution has its limitations, the precision of the estimation is bound by the size of the grid cells. A future direction will be to look at solutions that can estimate the (relative) coordinates of the sound source directly, without discretizing the spaces into grid cells.

9.3.4. Conclusion

In this section, we reported our literature study for sound source localization (SSL) methods, and underwater SSL datasets. We found no suitable dataset to train a deep learning model for the task of underwater SSL. This is due to the difficulty in data collect. We advise to use a simulator to generate a large scale of data to train a deep learning model for underwater SSL.

Finally, we showed our preliminary results in recreating the paper of Grinstein et al. [64], as well as additional experiments with a different loss function. We conclude that in the problem setting proposed by the author, a regression model with mean squared error as loss produces the lowest error for SSL. A qualitative analysis demonstrates the shortcoming of the grid-based SSL solution, which can be improved by developing a method that directly outputs the (relative) coordinates of the sound source without discretizing the spaces into grids.

10. Conclusions

In this document, we proposed a novel framework for feature-level fusion for the task of classifying domestic activities using a wireless acoustic sensor network (WASN) and a novel dataset that has sparse sound source and listeners in a large indoor environment.

Our framework utilizes a graph neural network (GNN) to perform the sensor fusion, which alleviates the shortcoming of the classifier stacking method on overfitting to input positions and only accepting a fixed number of sensor inputs. We introduced the message condensation layer to control the communication cost at the edge devices

and node dropout technique to reinforce the robustness of the framework against varying amount of sensor information to fuse.

We conducted different experiments on the novel dataset. As the power consumption of the sensors is dominated by the communication cost it incurs with the remote fusion center, we use the message length that the sensor must transmit to approximate their power consumption. We showed that our feature-level fusion framework does not increase the power consumption or computation cost for the acoustic sensors. Our framework achieves a better performance compared to decision-level fusion strategies implemented in prior work with a similar message length to transmit. With a reduced message length, our framework can perform on par with the decision-level fusion baselines. We also outlined the initial experimentation that has been done on extending the method to also allow sound-source localization.

Our research marks the initial step into using feature-level fusion for WASN applications. The novel developed algorithms can be utilized as reference implementations in the "OpenSwarm implementation" and within the proof of concepts (especially PoC3: Ocean Noise Pollution Monitoring).

Subsequent studies can be conducted to expand this concept further. These include:

1) Designing a dynamic message condensation process to let the edge device choose the message length based on its current power.
2) Embed the graph neural network fusion center to the edge devices, such that the sensor fusion process happens in a decentralized manner. This reduces the communication cost of the sensors.
3) Implement on-device continual learning such that the feature extraction and message condensation improves over time and can deal with data drifts.
4) Implement over-the-air computation techniques that aggregates wireless data during transmission to reduce the latency and computation cost for sensor fusion further.

11. References

[1] Y. Zhu *et al.*, 'A Survey on Deep Graph Generation: Methods and Applications'. *Proceedings of the First Learning on Graphs Conference*, PMLR 198:47:1-47:21, 2022

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, 'The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains', *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[3] X. Chen, 'Understanding spectral graph neural network', *arXiv preprint arXiv:2012.06660*, 2020.

[4] R. Zhu *et al.*, 'Aligraph: A comprehensive graph neural network platform', *arXiv preprint arXiv:1902.08730*, 2019.

[5] Y. Liu *et al.*, 'GraphTheta: A distributed graph neural network learning system with flexible training strategy', *arXiv preprint arXiv:2104.10569*, 2021.

[6] V. Md *et al.*, 'Distgnn: Scalable distributed training for large-scale graph neural networks', in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.

[7] D. Zheng *et al.*, 'DistDGL: Distributed graph neural network training for billion-scale graphs', in *2020 IEEE/ACM 10th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, IEEE, 2020, pp. 36–44.

[8] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, and H. Yu, 'Federated graph neural networks: Overview, techniques, and challenges', *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[9] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, and S. Avestimehr, 'Spreadgnn: Serverless multi-task federated learning for graph neural networks', *arXiv preprint arXiv:2106.02743*, 2021.

[10] Y. Pei *et al.*, 'Decentralized federated graph neural networks', in *International workshop on federated and transfer learning for data sparsity and confidentiality in conjunction with IJCAI*, 2021.

[11] Z. Gao, F. Gama, and A. Ribeiro, 'Wide and deep graph neural network with distributed online learning', *IEEE Transactions on Signal Processing*, vol. 70, pp. 3862–3877, 2022.

[12] I. Bistritz, A. Mann, and N. Bambos, 'Distributed distillation for on-device learning', *Advances in Neural Information Processing Systems*, vol. 33, pp. 22593–22604, 2020.

[13] G. Dekkers, F. Rosas, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, 'Dynamic sensor activation and decision-level fusion in wireless acoustic sensor networks for classification of domestic activities', *Information Fusion*, vol. 77, pp. 196–210, 2022.

[14] R. Grzeszick, A. Plinge, and G. A. Fink, 'Bag-of-features methods for acoustic event detection and classification', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1242–1252, 2017.

[15] J. Kürby, R. Grzeszick, A. Plinge, and G. A. Fink, 'Bag-of-Features Acoustic Event Detection for Sensor Networks.', in *DCASE*, 2016, pp. 55–59.

[16] J. Li, X. Zhang, Q. Zhou, F. T. Chan, and Z. Hu, 'A feature-level multi-sensor fusion approach for in-situ quality monitoring of selective laser melting', *Journal of Manufacturing Processes*, vol. 84, pp. 913–926, 2022.

[17] D. Lahat, T. Adali, and C. Jutten, 'Multimodal data fusion: an overview of methods, challenges, and prospects', *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[18] S. Chaib, H. Liu, Y. Gu, and H. Yao, 'Deep feature fusion for VHR remote sensing scene classification', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775–4784, 2017.

[19] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, 'A survey of decision fusion and feature fusion strategies for pattern classification', *IETE Technical review*, vol. 27, no. 4, pp. 293–307, 2010.

[20] A. N. Ahmed, A. Anwar, S. Mercelis, S. Latré, and P. Hellinckx, 'FF-GAT: Feature Fusion Using Graph Attention Networks', in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2021, pp. 1–6.

[21] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, 'Attentional feature fusion', in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3560–3569.

[22] B. V. Dasarathy, *Decision fusion*, vol. 1994. IEEE Computer Society Press Los Alamitos, 1994.

[23] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, 'A survey on machine learning for data fusion', *Information Fusion*, vol. 57, pp. 115–129, 2020.

[24] F. Castanedo and others, 'A review of data fusion techniques', *The scientific world journal*, vol. 2013, 2013.

[25] T. P. Banerjee and S. Das, 'Multi-sensor data fusion using support vector machine for motor fault detection', *Information Sciences*, vol. 217, pp. 96–107, 2012.

[26] K. Kolanowski, A. Świetlicka, R. Kapela, J. Pochmara, and A. Rybarczyk, 'Multisensor data fusion using Elman neural networks', *Applied Mathematics and Computation*, vol. 319, pp. 236–244, 2018.

[27] F. Rosas *et al.*, 'Optimizing the code rate of energy-constrained wireless communications with HARQ', *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 191–205, 2015.

[28] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, 'Sensor and sensor fusion technology in autonomous vehicles: A review', *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[29] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, 'Deep learning sensor fusion for autonomous vehicle perception and localization: A review', *Sensors*, vol. 20, no. 15, p. 4220, 2020.

[30] A. N. Ahmed, I. Ravijts, J. de Hoog, A. Anwar, S. Mercelis, and P. Hellinckx, 'A Joint Perception Scheme For Connected Vehicles', in *2022 IEEE Sensors*, IEEE, 2022, pp. 1–4.

[31] A. N. Ahmed, S. Mercelis, and A. Anwar, 'Graph Attention Based Feature Fusion For Collaborative Perception', in *2024 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2024, pp. 2317–2324.

[32] O. Kreibich, J. Neuzil, and R. Smid, 'Quality-based multiple-sensor fusion in an industrial wireless sensor network for MCM', *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4903–4911, 2013.

[33] B. P. L. Lau *et al.*, 'A survey of data fusion in smart city applications', *Information Fusion*, vol. 52, pp. 357–374, 2019.

[34] B. P. L. Lau, N. Wijerathne, B. K. K. Ng, and C. Yuen, 'Sensor fusion for public space utilization monitoring in a smart city', *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 473–481, 2017.

[35] J. Sui, T. Adali, Q. Yu, J. Chen, and V. D. Calhoun, 'A review of multivariate methods for multimodal fusion of brain imaging data', *Journal of neuroscience methods*, vol. 204, no. 1, pp. 68–81, 2012.

[36] A. Mesaros *et al.*, 'DCASE 2017 challenge setup: Tasks, datasets and baseline system', in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[37] L. Cristoforetti *et al.*, 'The DIRHA simulated corpus', in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds., Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 2629–2634. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/650_Paper.pdf

[38] A. Temko, D. Macho, C. Nadeu, and C. Segura, 'UPC-TALP database of isolated acoustic events', *Internal UPC report*, vol. 85, 2005.

[39] G. Dekkers *et al.*, 'The SINS database for detection of daily activities in a home environment using an acoustic sensor network', *Detection and Classification of Acoustic Scenes and Events 2017*, pp. 1–5, 2017.

[40] M. Vacher *et al.*, 'The sweet-home project: Audio technology in smart homes to improve well-being and reliance', in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2011, pp. 5291–5294.

[41] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos, 'Multi-microphone fusion for detection of speech and acoustic events in smart spaces', in *2014 22nd European Signal Processing Conference (EUSIPCO)*, IEEE, 2014, pp. 2375–2379.

[42] I. Martín-Morató, M. Cobos, and F. J. Ferri, 'Analysis of data fusion techniques for multi-microphone audio event detection in adverse environments', in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2017, pp. 1–6.

[43] X. Xu, Z. Tao, W. Ming, Q. An, and M. Chen, 'Intelligent monitoring and diagnostics using a novel integrated model based on deep learning and multi-sensor feature fusion', *Measurement*, vol. 165, p. 108086, 2020.

[44] X. Zhang, X. Zhang, J. Liu, B. Wu, and Y. Hu, 'Graph features dynamic fusion learning driven by multi-head attention for large rotating machinery fault diagnosis with multi-sensor data', *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106601, 2023.

[45] Manolis Savva* *et al.*, 'Habitat: A Platform for Embodied AI Research', in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[46] C. Chen *et al.*, 'SoundSpaces 2.0: A Simulation Platform for Visual-Acoustic Learning', in *NeurIPS 2022 Datasets and Benchmarks Track*, 2022.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 'Dropout: a simple way to prevent neural networks from overfitting', *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] A. Bertrand, 'Applications and trends in wireless acoustic sensor networks: A signal processing perspective', in *2011 18th IEEE symposium on communications and vehicular technology in the Benelux (SCVT)*, IEEE, 2011, pp. 1–6.

[49] F. Alías, R. M. Alsina-Pagès, and others, 'Review of wireless acoustic sensor networks for environmental noise monitoring in smart cities', *Journal of sensors*, vol. 2019, 2019.

[50] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, 'Detection and classification of acoustic scenes and events', *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[51] A. Mesaros *et al.*, 'Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2017.

[52] Y. Bai, L. Lu, J. Cheng, J. Liu, Y. Chen, and J. Yu, 'Acoustic-based sensing and applications: A survey', *Computer Networks*, vol. 181, p. 107447, 2020.

[53] A. Ståhlbröst, A. Sällström, and D. Hollosi, 'Audio monitoring in Smart Cities: an information privacy perspective', 2014.

[54] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, 'Graph attention networks', *arXiv preprint arXiv:1710.10903*, 2017.

[55] I. Loshchilov and F. Hutter, 'Decoupled weight decay regularization', *arXiv preprint arXiv:1711.05101*, 2017.

[56] K. He, X. Zhang, S. Ren, and J. Sun, 'Delving deep into rectifiers: Surpassing human-level performance on imagenet classification', in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[57] A. Agarap, 'Deep Learning Using Rectified Linear Units (ReLU)', *arXiv preprint arXiv:1803.08375*, 2018.

[58] Q. Li, Z. Han, and X.-M. Wu, 'Deeper insights into graph convolutional networks for semi-supervised learning', in *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[59] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, 'Graph matching networks for learning the similarity of graph structured objects', in *International conference on machine learning*, PMLR, 2019, pp. 3835–3845.

[60] M. Cobos, F. Antonacci, A. Alexandridis, A. Mouchtaris, and B. Lee, 'A survey of sound source localization methods in wireless acoustic sensor networks', *Wireless Communications and Mobile Computing*, vol. 2017, no. 1, p. 3956282, 2017.

[61] F. Thomas and L. Ros, 'Revisiting trilateration for robot localization', *IEEE Transactions on robotics*, vol. 21, no. 1, pp. 93–101, 2005.

[62] F. Gustafsson and F. Gunnarsson, 'Positioning using time-difference of arrival measurements', in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, IEEE, 2003, p. VI–553.

[63] P. Pertilä, T. Korhonen, and A. Visa, 'Measurement combination for acoustic source localization in a room environment', *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2008, pp. 1–14, 2008.

[64] E. Grinstein, M. Brookes, and P. A. Naylor, 'Graph neural networks for sound source localization on distributed microphone networks', in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[65] S. Argentieri and P. Danes, 'Broadband variations of the MUSIC high-resolution method for sound source localization in robotics', in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 2009–2014.

[66] S. Astapov, J. Berdnikova, and J.-S. Preden, 'Optimized acoustic localization with SRP-PHAT for monitoring in distributed sensor networks', *International Journal of Electronics and Telecommunications*, 2013.

[67] A. Santoro *et al.*, 'A simple neural network module for relational reasoning', *Advances in neural information processing systems*, vol. 30, 2017.

[68] C. Knapp and G. Carter, 'The generalized correlation method for estimation of time delay', *IEEE transactions on acoustics, speech, and signal processing*, vol. 24, no. 4, pp. 320–327, 1976.

[69] L. Wilson, M. K. Pine, and C. A. Radford, 'Small recreational boats: A ubiquitous source of sound pollution in shallow coastal habitats', *Marine Pollution Bulletin*, vol. 174, p. 113295, 2022.

[70] L. Wilson, R. Constantine, T. van der Boon, and C. A. Radford, 'Using timelapse cameras and machine learning to enhance acoustic monitoring of small boat sound', *Ecological Indicators*, vol. 142, p. 109182, 2022.

[71] D. Santos-Domínguez, S. Torres-Guijarro, A. Cardenal-López, and A. Pena-Gimenez, 'ShipsEar: An underwater vessel noise database', *Applied Acoustics*, vol. 113, pp. 64–69, 2016.

[72] M. Irfan, Z. Jiangbin, S. Ali, M. Iqbal, Z. Masood, and U. Hamid, 'DeepShip: An underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification', *Expert Systems with Applications*, vol. 183, p. 115270, 2021.

[73] L. Hatch *et al.*, 'SanctSound: Building Data Systems for Sound Decisions', in *The Effects of Noise on Aquatic Life: Principles and Practical Considerations*, Springer, 2024, pp. 1–11.

[74] P. Jin, B. Wang, L. Li, P. Chao, and F. Xie, 'Semi-supervised underwater acoustic source localization based on residual convolutional autoencoder', *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 107, 2022.

[75] D. Qin, J. Tang, and Z. Yan, 'Underwater acoustic source localization using LSTM neural network', in *2020 39th Chinese Control Conference (CCC)*, IEEE, 2020, pp. 7452–7457.

[76] R. Lefort, G. Real, and A. Drémeau, 'Direct regressions for underwater acoustic source localization in fluctuating oceans', *Applied Acoustics*, vol. 116, pp. 303–310, 2017.

[77] Y. Le and X. Yang, 'Tiny imagenet visual recognition challenge', *CS 231N*, vol. 7, no. 7, p. 3, 2015.