

Article

Towards a Gated Graph Neural Network with an Attention Mechanism for Audio Features with a Situation Awareness Application

Jieli Chen ^{1,2}, Kah Phooi Seng ^{3,4,*}, Li Minn Ang ⁴, Jeremy Smith ² and Hanyue Xu ^{1,2} 

¹ School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China; jieli.chen22@student.xjtlu.edu.cn (J.C.); hanyue.xu19@student.xjtlu.edu.cn (H.X.)

² Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool L69 3BX, UK; j.s.smith@liverpool.ac.uk

³ Faculty of AI and Frontier Technology, UNITAR International University, Petaling Jaya 47301, Malaysia

⁴ School of Science, Technology and Engineering, University of Sunshine Coast, Petrie, QLD 4502, Australia; lang@usc.edu.au

* Correspondence: jasmine.seng@unitar.my or jseng@usc.edu.au

Abstract

Situation awareness (SA) involves analyzing sensory data, such as audio signals, to identify anomalies. While acoustic features are widely used in audio analysis, existing methods face critical limitations; they often overlook the relevance of SA audio segments, failing to capture the complex relational patterns in audio data that are essential for SA. In this study, we first propose a graph neural network (GNN) with an attention mechanism that models SA audio features through graph structures, capturing both node attributes and their relationships for richer representations than traditional methods. Our analysis identifies suitable audio feature combinations and graph constructions for SA tasks. Building on this, we introduce a situation awareness gated-attention GNN (SAGA-GNN), which dynamically filters irrelevant nodes through max-relevance neighbor sampling to reduce redundant connections, and a learnable edge gated-attention mechanism that suppresses noise while amplifying critical events. The proposed method employs sigmoid-activated attention weights conditioned on both node features and temporal relationships, enabling adaptive node emphasizing for different acoustic environments. Experiments reveal that the proposed graph-based audio features demonstrate superior representation capacity compared to traditional methods. Additionally, both proposed graph-based methods outperform existing approaches. Specifically, owing to the combination of graph-based audio features and dynamic selection of audio nodes based on gated-attention, SAGA-GNN achieved superior results on two real datasets. This work underscores the importance and potential value of graph-based audio features and attention mechanism-based GNNs, particularly in situational awareness applications.

Keywords: situation awareness; graph neural networks; attention mechanism; anomalous sound; audio classification



Received: 13 May 2025
Revised: 11 June 2025
Accepted: 20 June 2025
Published: 28 June 2025

Citation: Chen, J.; Seng, K.P.; Ang, L.M.; Smith, J.; Xu, H. Towards a Gated Graph Neural Network with an Attention Mechanism for Audio Features with a Situation Awareness Application. *Electronics* **2025**, *14*, 2621. <https://doi.org/10.3390/electronics14132621>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Situation awareness (SA) refers to the process of perceiving, comprehending, and projecting elements in an environment [1], particularly in scenarios requiring real-time responses, such as disaster management [2,3], security systems [4], and anomaly detection [5]. With audio signals serving as one of the primary sensory modalities in SA tasks, analyzing

their features effectively is crucial. However, with the growing complexity of audio signal processing in SA tasks, conventional methods relying solely on individual audio features and traditional machine learning techniques are often inadequate for capturing intricate temporal and structural patterns within audio data [6,7]. Traditional approaches for audio signal analysis often rely on single features such as spectral or cepstral representations, alongside classifiers such as convolutional neural networks (CNNs) [5,8]. While effective in isolated scenarios, these methods are limited in their ability to capture the complex temporal dependencies and relational patterns inherent in audio data, particularly for SA applications where context and structure are critical [9].

Traditional audio feature extraction methods, such as Mel-frequency cepstral coefficients (MFCC) [10] and spectrograms, are widely utilized and well-researched in audio classification and anomaly detection. For example, log-Mel spectrograms [10] and constant-Q transform (CQT) [11] have shown promising results in tasks involving environmental sound classification [12,13] and music genre recognition [14] due to their ability, for example, to emphasize perceptually relevant frequency components that align with human auditory perception and resolve fine harmonic structures [15]. In SA applications, these audio features can provide relevant clues to SA using frequency information (e.g., different audio times may have different frequency intervals) and temporal dynamics (e.g., faulty sounds that appear periodically in anomaly detection) [13,16], which are critical for perceiving and comprehending environmental elements. However, these features are often modeled independently or spliced together, without adequately addressing their interdependencies and individual task-specific contributions [17,18]. Existing methods, predominantly based on CNNs [5,19,20], often fail to consider the complex relationships between audio segments when representing audio features.

Graph structures provide a framework for addressing these limitations by modeling relationships and dependencies within data. Graph neural networks (GNNs) enable the representation of structured data and the extraction of relational patterns through message-passing mechanisms. These models have significantly advanced fields such as computer vision [21], natural language processing [22], and decision-making systems [23]. In the audio domain, graph structures offer a way to represent both the features of audio frames and their interdependencies as nodes and edges, capturing richer contextual information than traditional methods [24,25]. In the field of SA, there have also been some studies that have proposed the use of graph structures to represent audio for better modeling of the temporal dependencies and complexity patterns inherently exhibited in audio data [26,27], which has achieved a better performance than traditional approaches on a number of public datasets. Despite the potential of graph methods for audio processing, the applicability of different graph structures to different audio features and combinations, especially in SA tasks, remains relatively unexplored. Furthermore, recent works in the field of SA audio have highlighted the challenge that SA audio—as a subset of general audio—faces, characterized by a preponderance of noise segments and a scarcity of critical segments [28].

Attention mechanisms further enhance the adaptability of GNNs by allowing models to focus on the most relevant aspects of the input data. Attention-based models, such as graph attention networks (GAT) [29] and graph transformers [30,31], have demonstrated substantial improvements in performance by adaptively weighting connections between nodes. These advancements align closely with the demands of SA tasks, where the ability to capture contextual and structural information is crucial [32,33].

To address these gaps, we investigate the graph-based method with a situation awareness audio attention mechanism and its features. The structure is illustrated in Figure 1. The proposed method is experimentally demonstrated to obtain superior representation, as

well as classification results, compared to traditional methods in SA-related real datasets. Our contributions are threefold:

- We explore the performance of different audio features in SA tasks and propose audio features and their combinations suitable for graph modeling. With this approach, we uncovered the potential of graph-based features for SA applications, which can lead to more discriminative characterizations compared to traditional methods.
- We propose a method by which to convert audio features into graph representations, enabling the model to capture the relational patterns inherent in audio data. By converting specific audio features into different graph structures, e.g., connecting nodes representing audio frames in chronological order or according to the similarity of audio frames, we explore the impact of different graph structures on the model's performance in SA. This process enables the model to utilize the rich information in the audio signal, such as temporal dependencies.
- We introduce two graph-based models for situational awareness in audio analysis. First, we propose an attention-enhanced graph neural network that employs neighborhood attention to dynamically weight node relationships during feature aggregation. Building upon this foundation, our second and primary contribution is the situation awareness gated-attention GNN (SAGA-GNN), which significantly advances the architecture through two novel mechanisms: max-relative node sampling that selectively retains only the most task-relevant connections, and a learnable gating system that adaptively filters noisy edges while amplifying critical audio event features. The SAGA-GNN demonstrates superior performance by explicitly addressing the unique challenges of SA audio graphs, where noise and irrelevant acoustic events typically compromise traditional graph representations.

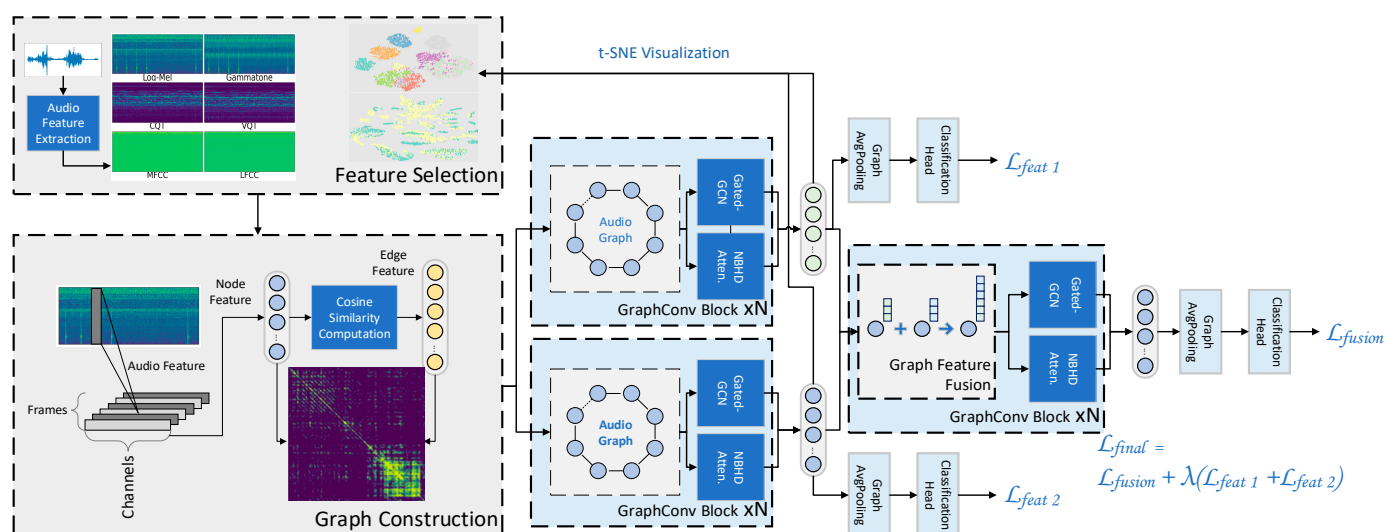


Figure 1. Proposed GNN with neighborhood attention for audio features of situation awareness. Gated-GCN: gated graph convolutional network; NBHD Atten: neighborhood attention.

2. Related Work

2.1. Audio-Based Situation Awareness

In audio-based SA, deep learning-based methods are the current mainstream [8,19,20,34]. For instance, in [35], the authors compared the performance of CNNs and long short-term memory (LSTM) networks using Mel-frequency cepstral coefficients (MFCC) as audio features for temporal classification tasks in natural disaster scenarios. Their results demonstrated that CNNs slightly outperformed LSTMs, highlighting the effectiveness of CNNs in recognizing intricate patterns in audio features. This work underscores the importance

of leveraging advanced neural architectures for audio-based SA tasks. Further advancements have been made at the audio feature level, where researchers have explored different methods to enhance the representation and analysis of audio signals. For example, the authors in [36] proposed a feature-level anomaly detection approach by learning the discrete distribution of audio features to identify anomalous sounds. Similarly, in [5], a hybrid method was introduced that combines frequency-domain features (log-Mel spectrograms) with CNN-based time-domain features for audio anomaly detection. This approach leverages the complementary information from different audio features to create a more robust representation of the audio signal, demonstrating the value of integrating multiple feature types for improved performance.

While these methods have shown promise, they fall short in the way they capture the complex interdependencies and relational patterns inherent in audio data. This limitation has motivated the exploration of graph-based approaches, which offer a more natural and effective way to model the structural relationships within audio signals. Recent studies have demonstrated the superior performance of graph-based methods in audio processing tasks. For example, graph neural networks (GNNs) have been shown to outperform traditional CNNs and LSTMs in tasks such as sound event detection, audio classification, and anomaly detection [26,27,37].

2.2. Graph-Based Methods for Audio Processing

Some recent work has taken advantage of the ability of GNNs to capture complex relationships in data by converting audio to a graph structure and processing it using graph-based methods. There are several approaches to converting audio into graph structures. In [38], the authors used a pre-trained CNN to map log-Mel spectrograms into embeddings representing audio events and used these embeddings as nodes to construct a graph to model the relationships between different sound events. Another approach for constructing audio graphs is based on the inherent structure of audio features. For example, following the frame-to-node transformation, this construction method mainly utilizes the temporal correlation between audio frames [39]. In [40], a way of patching the spectrogram is used, where each region is treated as a node to represent the correlation of sound events in different regions of the spectrogram. Several works have explored the impact of graph structure on audio processing. In [41], the authors propose learnable graph structures that incorporate the adjacency matrix of the audio graph into learnable parameters in the model to optimize the graph structure during training. This approach also assumes that nodes that are temporally close have stronger relationships. In [24], the authors propose an audio graph construction method based on clustering techniques and combined with GNN for audio classification tasks. The audio graph constructed using k nearest neighbors (kNN) and fuzzy C-means clustering algorithms exploits local and higher-order relationships between audio frames, respectively.

These related works demonstrate the rationality and effectiveness of graph-based approaches for audio processing, but few works have explored the applicability of different audio features and their combinations in the graph learning framework.

2.3. GNNs with Attention Mechanisms

Gong et al. first applied the transformer architecture to audio classification by proposing the audio spectrogram transformer (AST) [42]. The global context is obtained from the log-Mel spectrogram segmented into patches by the self-attention mechanism. Experiments show that the self-attention mechanism can learn a superior audio representation compared to the CNN architecture. The authors in [43] use GAT as a component for learning temporal dependencies within audio features to obtain more efficient audio representations and utilize Transformer to predict audio captions from the representations. In [44], the authors connect graph convolutional network (GCN) and transformer in order to be

responsible for learning spatial features and temporal contextual information in the audio graph, respectively. The authors in [45] experimentally demonstrate that in a hybrid GNN and self-attention layer, the GNN can learn the inherent relationships (i.e., edge features) between nodes, and the self-attention layer can implicitly utilize these features to further infer the contextual relationships between nodes. However, there is a lack of research on attention mechanisms for audio graphs in the context of SA audio.

3. Methodology

3.1. Graph Construction for Audio

Audio modalities are not inherently represented as graph structures because they are typically modeled as time-series data or spectrograms, which are grid-like representations in the time–frequency domain [40]. Therefore, these modalities must be manually transformed into graph structures. Constructing graph structure from the acoustic features, a sliding window mechanism is designed. Given an acoustic feature of size $(F \times T)$, where F represents the number of frequency bins and T represents the number of time steps t (or frames), a set of feature vectors $X = \{x_1, x_2, \dots, x_T\}$ is obtained by convolving it with a sliding window using a two-dimensional convolutional layer with kernel size of $(F \times n)$, with n representing the width in the time dimension for convolution. Due to the nature of the sliding convolution, each feature vector represents a specific range of frames $\left(t - \frac{n-1}{2}, t + \frac{n-1}{2}\right)$ in the acoustic signature without considering padding on sides.

In this approach, we treat each frame of the acoustic feature as a node in a graph. Specifically, let $v_t \in V$ represent the node corresponding to the frame at time t , where V is the set of all nodes. These nodes are then connected by edges according to predefined rules. For example, one simple rule could be to connect each node to the next in a sequence (i.e., connect v_t and v_{t+1}), creating edges that represent temporal relationships between consecutive frames. This set of nodes and edges forms a frame graph $G = (V, E)$, where V is the set of nodes (frames) and E is the set of edges (connections between frames).

3.2. Graph Structure for Audio

Since audio is inherently a temporal signal, it is natural to model its structure using graph-based representations, where the segmented audio frames can be treated as nodes. In recent work, the effectiveness of representing audio as a graph has been demonstrated, particularly in the association of temporal sequence information within audio [46] and the relationships between audio events [38]. When considering the associations within time series, each audio frame can be represented as a node, and these nodes are connected sequentially to capture the temporal dependencies between consecutive frames. Figure 2 intuitively illustrates the adjacency matrices using different node connection methods. For a line graph structure (Figure 2a), where edges between nodes represent the temporal progression of the audio signal. Further, by connecting the front and back nodes, a cycle graph structure can be formed (Figure 2b). Such structures are particularly useful for tasks like speech recognition, audio classification, and sound event detection, as it preserves the sequential nature of the data while allowing for the application of graph-based algorithms [37].

For an audio graph that contains N nodes, the adjacency matrices of line graph and cycle graph can be expressed as follows:

$$A_{i,j}^{line} = \begin{cases} 1 & \text{if } |i - j| = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$A_{i,j}^{cycle} = \begin{cases} 1 & \text{if } |i - j| = 1 \text{ or } |i - j| = N - 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $A_{i,j}^{line}$ and $A_{i,j}^{cycle}$ denote the connection of a source node with node index i to a target node with node index j in the line graph and cyclic graph, respectively.

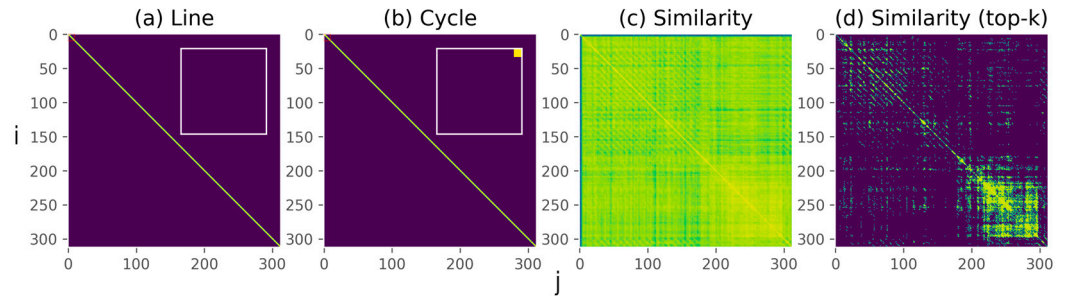


Figure 2. Adjacency matrixes of different graph construction methods.

Dynamic graph structures have been proven to enhance the data representation capabilities of graph learning architectures across various domains [47]. Some work has suggested a dynamic approach to graph construction when composing graphs of audio to take into account the inherent evolutionary patterns of audio data [39,41]. Therefore, a similarity map can be obtained by calculating the similarity between pairs of nodes to be used as an adjacency matrix for constructing the graph (Figure 2c,d). Specifically, the following equations express the similarity matrix by calculating cosine similarity of graph nodes feature:

$$A_{i,j}^{Simi} = g(f(x_i, x_j)) \quad (3)$$

where $A_{i,j}^{Simi}$ denotes the similarity value between the i^{th} and j^{th} nodes, x_i denotes the i^{th} node feature, and $g(\cdot)$ is the normalization function. $f(\cdot)$ computes the similarity of a pair of features, which can be expressed as follows:

$$f(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \cdot \|x_j\|} \quad (4)$$

where $\|\cdot\|$ denotes the Euclidean norm. To avoid constructing a graph that is a fully connected graph (the computation of graph convolution is proportional to the number of nodes and the number of edges), top k, as a threshold, is applied to each graph construction process:

$$\bar{A}_{i,j}^{Simi} = \begin{cases} A_{i,j}^{Simi} & \text{if } A_{i,j}^{Simi} \in \text{Top}_k(A_{i,j}^{Simi}) \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

$$\text{Top}_k(A_{i,j}^{Simi}) = \{A_1^{Simi}, A_2^{Simi}, \dots, A_k^{Simi}\} \quad (6)$$

where $\bar{A}_{i,j}^{Simi}$ denotes the value from source node i to target node j in the final adjacency matrix, and $\text{Top}_k(\cdot)$ denotes the top k algorithm.

3.3. GNN with Attention Mechanisms

We propose a graph-based model for SA audio, which is a hybrid GNN and audio-specific attention mechanisms (Figure 1). Inspired by GraphGPS [30], an architecture that parallelizes MPNNs with global attention is used, aiming for message passing in a way that both respects the graph structure and provides global attention.

Based on the discussion of graph construction methods, the enhancement of similarity graphs over line or cycle graphs can be explained by the fact that line or cycle graphs provide short-distance connections for audio nodes, while similarity graphs provide reasonably long-distance connections while retaining short-distance connections. The difference between the graph models, on the other hand, lies in how neighboring nodes are aggre-

gated and how the features of the original nodes are updated. Based on the gated graph convolutional network (Gated-GCN), the message passing for the l^{th} layer f_{G-GCN}^l can be expressed as [48] follows:

$$h_i^{l+1} = f_{G-GCN}^l(h_i^l, \{h_j^l : j \rightarrow i\}) = \sigma\left(U^l h_i^l + \sum_{j \rightarrow i} \phi_{ij}^l \otimes V^l h_j^l\right), \quad (7)$$

$$\phi_{ij}^l = \sigma(\bar{U}^l h_i^l + \bar{V}^l h_j^l), \quad (8)$$

where f_{GCN}^l denotes the l^{th} GCN layer; U and V are learnable weights; h_i^l is the feature vector of the i^{th} node after the l^{th} layer's graph convolution; and σ represents the nonlinear activation function, which is practically a rectified linear unit. The learnable gate ϕ_{ij} for the edge e_{ij} between node h_i and h_j allows the network to regulate the rate at which the nodes' information is aggregated. This is essentially computing the weights of the edges between nodes h_i and h_j from their features and perform weighted message passing on nodes.

However, even if different graph structures are constructed for different audio samples using similarity graphs, connections based on similarity alone may ignore the connections of individual nodes. For example, one possible scenario is that in an audio clip that is silent for an extended period of time or has a noisy background sound, the audio events are mainly represented by individual frame nodes. In this case, the similarity $A_{a,b}^{simi}$ will be higher than the similarity $A_{a,c}^{simi}$, where a and b are in the set of event-independent node indices, $I_{independent}$, and c is in the set of event-related node indices, $I_{related}$. According to the message passing in GNN, this may result in the event-related node $V_{related} = \{v_r | r \in I_{related}\}$ having less weight compared to the other nodes when performing the message passing, causing a lesser contribution of $V_{related}$ to the audio representation.

Introducing an attention mechanism is a way to capture the long-range context [42], which can provide global information based on the graph structure [30], which is particularly critical in SA tasks where events or anomalies may be influenced by non-local contextual information [24]. The attention mechanism computes a point-to-point attention map, assigning weights to node pairs based on their relevance to the task. This ensures that the model focuses on the most salient features and their relationships in audio signals [42]. Specifically, the scaled dot-product attention, widely adopted in Transformer architectures [49], is employed to enhance the graph's ability to integrate global feature interactions efficiently. It can be represented as follows:

$$H_{Attn}^l = \text{softmax}\left(\text{AttnMap}^l\right) H_v^l \quad (9)$$

$$\text{AttnMap}^l = \frac{H_q^l H_k^{lT}}{\sqrt{d_k}} \quad (10)$$

$$H_q^l = QH^l, H_k^l = KH^l, H_v^l = VH^l \quad (11)$$

where H^l denotes the set of nodes feature of the l^{th} graph convolutional layer. Q , K , and V are learnable weights, representing query, index, and content, respectively, and are optimized by minimizing a loss function required by the task, e.g., cross-entropy loss in this study (discussed in Section 3.3.2). In this work, since they are all obtained based on the same set of node features H^l , H_q^l and H_k^l represent the information about what a node is "looking for" and "offering" in its neighbor nodes [49]. The AttnMap^l obtained by their proportional product in Equation (10) can be viewed as a weighted adjacency matrix from the point of view of the graph structure, where each value $\text{AttnMap}_{i,j}^l$ represents the attention value of the source node i for the target node j . In contrast to binary or fixed-edge

weights (as in traditional graphs), the weights are dynamically computed based on the node features and learned transformations.

The role of global attention in our framework is clearly demonstrated in Figure 3, which reveals important distinctions between similarity-based and attention-based node connections. The similarity matrix $\bar{A}^{simi}$ establishes connections between nodes purely based on frame-level feature similarity, while the attention map shows a more nuanced pattern of connectivity that incorporates additional contextual information. In Figure 3a, we observe that nodes receiving larger attention weights correspond to frames with higher amplitude in the original waveform (particularly between frames 0 and 60), despite these frames showing relatively sparse connections in the similarity matrix. This discrepancy suggests that the attention mechanism is capturing important signal characteristics that simple similarity measures might miss. Further examination of Figure 3b,c reveals that the attention weights are strategically concentrated at specific nodes that complement the similarity matrix connections. For instance, frames 0–50 in Figure 3b and frames 150–200 in Figure 3c show particularly strong attention weights, even though these regions correspond to sustained sounds in the original waveform. Rather than just responding to signal energy, the attention weights appear to be learning meaningful relationships between nodes based on the underlying audio events and their contextual significance. The attention map's ability to identify and emphasize these event-related nodes demonstrates its capacity to extract features that are crucial for accurate audio analysis. This analysis suggests that our global attention mechanism effectively identified and emphasized nodes that correspond to meaningful audio events rather than just matching low-level features. The complementary nature of these two connection methods, with similarity capturing local feature relationships and attention highlighting event-related nodes, creates a more robust and informative graph representation for audio analysis tasks.

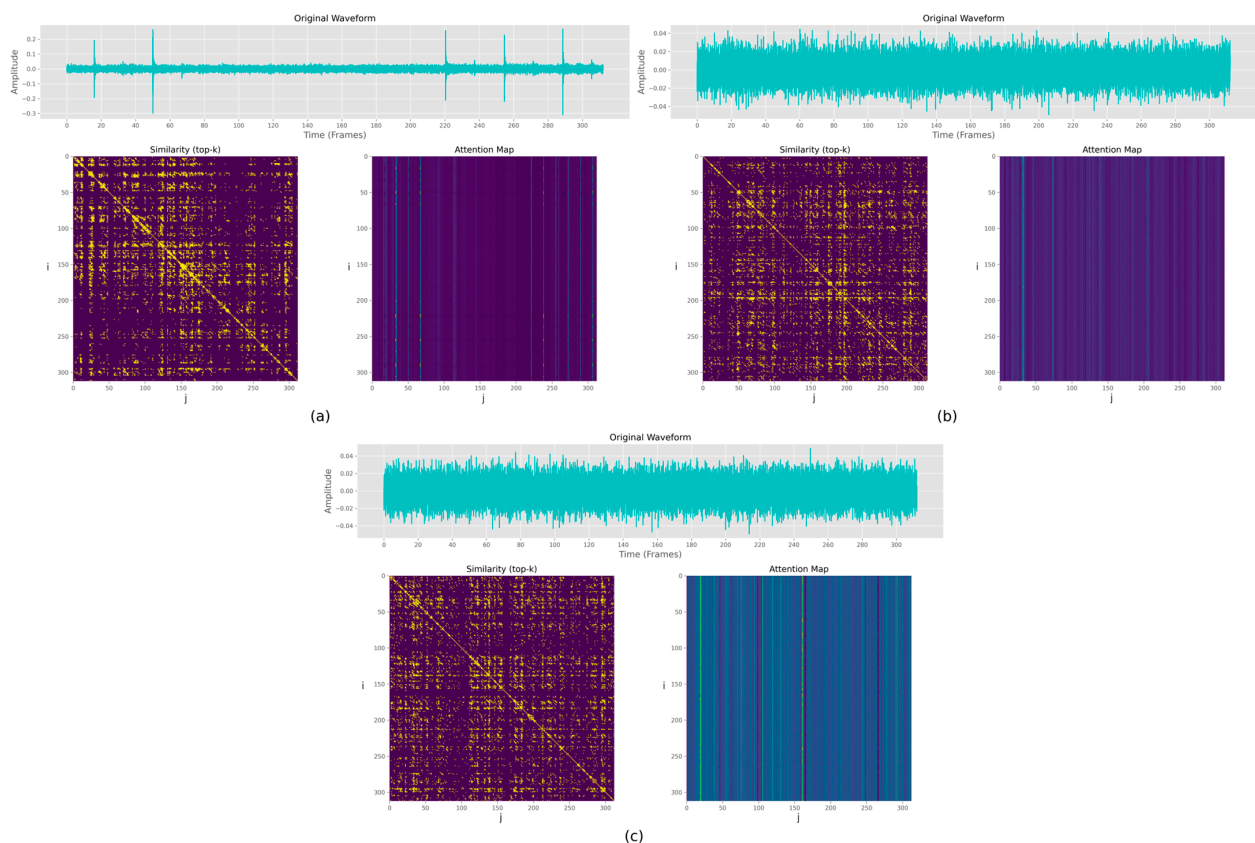


Figure 3. Similarity matrices compared with attention maps on samples. (a–c) illustrate different audio samples.

3.3.1. Neighborhood Attention Mechanism

On a basis that provides global attention, neighborhood attention [50] introduces an inductive bias inspired by convolution operations, which encourages each node in the feature set to focus primarily on its neighboring nodes. This bias is motivated by the idea that local neighborhood information, similar to how convolutions operate over local clips in audios or patches in images, is essential for capturing relational structures in graph data [51]. In the audio domain, a similar approach is to add clustering techniques in front of the standard Transformer, but this can lead to problems of slow convergence [24]. In practice, while the self-attention mechanism can attend to meaningful nodes, the connectivity of these nodes is usually global; i.e., each node is connected to all other nodes in the graph, regardless of their spatial or temporal proximity (e.g., attention maps shown in Figure 3). This may lead the attention layer to overly prospective or retrospective nodes [52], which cause the output elements of the layer to be overly dependent on any input elements from the “past” or “future” to the extent that it has a negative impact on the model performance [53]. In response to this problem, neighborly attention can be focused on a specific range to eliminate this possible dependency.

Given the temporal relevance that the audio graph focuses on. The neighborhood attention mechanism can focus on only a specific range of node i 's neighborhood to provide attention weights based on Equations (9)–(11). Using $N(i, j)$ to denote the j^{th} neighboring node of node i , the neighborhood attention can be expressed as follows [50]:

$$H_{N_Attn}^l(H_{i,j}^l) = \text{softmax} \left(\frac{H_{q_{i,j}}^l H_{k_{N(i,j)}}^{lT} + B_{i,j}}{\sqrt{d_k}} \right) H_{v_{N(i,j)}}^l \quad (12)$$

where $H_{N_Attn}^l(H_{i,j}^l)$ denotes the output of the l^{th} neighborhood attention layer for the i^{th} node's j^{th} neighboring node, and $B_{i,j}$ denotes the relative positional biases. In contrast to the standard transformer attention described in Equations (9)–(11), where the attention map is obtained through the complete Q , K , and V , neighborhood attention computes only the values in the range $N(i, j)$. This allows the neighbor attention layer to focus its attention on and incorporates relative positional bias similar to that of the convolutional layer [50].

Integrating Gated-GCN with neighborhood attention mechanisms, the proposed GNN with a neighborhood attention mechanism can be expressed as follows:

$$h_i^{l+1} = f_{G-GCN}^l \left(h_i^l, \left\{ h_j^l : j \rightarrow i \right\} \right) = \text{GELU} \left(U^l h_i^l + \sum_{j \rightarrow i} \phi_{ij}^l \otimes V^l h_j^l \right) + H_{N_Attn}^l \quad (13)$$

where ϕ_{ij}^l denotes the local attention along the edges in a similarity graph, and $H_{N_Attn}^l$ denotes the global attention without regard to graph structure.

3.3.2. Audio Feature Fusion

In order to improve the drawback of the traditional audio feature fusion methods, i.e., splicing audio features directly without considering their contribution to the task, the model employs a graph-based fusion after obtaining separate representations of the different features. Since different features are computed with the same audio clip, each of their nodes corresponds to the same frame. We splice graph nodes from different audio features at the feature dimension level:

$$H_{fusion} = \left[H_{feat1}, H_{feat2} \right] \in \mathbb{R}^{n \times (d_{feat1} + d_{feat2})} \quad (14)$$

where $H_{feat\ n}$ and H_{fusion} denote the nodes feature of the n^{th} audio feature and the fused feature, respectively.

From previous research, it is evident that adopting multi-level losses in graph learning architectures enables the capture of contextual information at the local (lower) levels and models node relationships at the global (higher) levels. Consequently, this work employs a supervised learning approach that sequentially trains both individual feature layers and the merged layers [54]. To ensure that the model learns sufficiently both before and after feature fusion, we compute and sum the cross-entropy loss for the features before and after fusion, respectively, which can be expressed as follows:

$$L_{final} = L_{fusion} + \lambda (L_{feat1} + L_{feat2}), \quad (15)$$

$$\text{with } L = -\log\left(\frac{e^{p_{g_i}}}{\sum_j e^{p_j}}\right)$$

where λ denotes the loss weight of the individual audio feature contributing to the loss function, p_{g_i} represents the i^{th} output logits, and p_j represents the j^{th} element of the logits. Cross-entropy loss is used in this case because it is well-suited for classification tasks as it quantifies the difference between predicted probabilities and the true class labels [55]. By applying this loss both before and after fusion, we ensure that the model effectively learns to distinguish between classes at multiple stages of feature processing [54].

4. Proposed Gated-Attention Graph Convolutional Network

The application of graph neural networks to situational awareness (SA) audio processing introduces unique computational challenges that demand specialized architectural solutions. Unlike general audio, SA audios contain unpredictable noise and brief critical events that require special handling. When we represent these recordings as graphs, where nodes are audio features and edges represent relationships, two key problems emerge: First, background noise creates misleading connections. Imagine a security microphone picking up both a gunshot (important) and constant wind noise (irrelevant). Traditional graph networks treat all connections equally, causing the wind noise to overwhelm the gunshot features after just a few processing layers. Traditional GCNs exacerbate this through isotropic smoothing [51], for example, mean aggregation:

$$h_i^{l+1} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} h_j^l \quad (16)$$

Second, noise or silent periods distort the graph. During quiet segments, all nodes look similarly “flat”, forming strong but meaningless connections between them.

The challenge of detecting critical but sparse audio events—such as screams, glass breaking, or gunshots—is particularly acute in SA applications, where these signals often manifest as small, isolated patterns in noisy environments, as we discussed in Section 3. Traditional graph convolutional networks (GCNs) struggle with such patterns because their uniform node-processing approach inadvertently propagates noise from irrelevant features while diluting critical but sparse signals. To address this, we propose SAGA-GNN (Figure 4), a graph-based model that explicitly enhances SA-relevant audio features through two key mechanisms: max-relative node sampling prioritizes nodes containing salient event signatures (e.g., abrupt onsets in gunshot spectrograms), while gated-attention edge updates dynamically suppress noisy connections. Our approach diverges from conventional GCNs by amplifying features of diagnostically valuable nodes—precisely the ones most likely to be overlooked in SA scenarios.

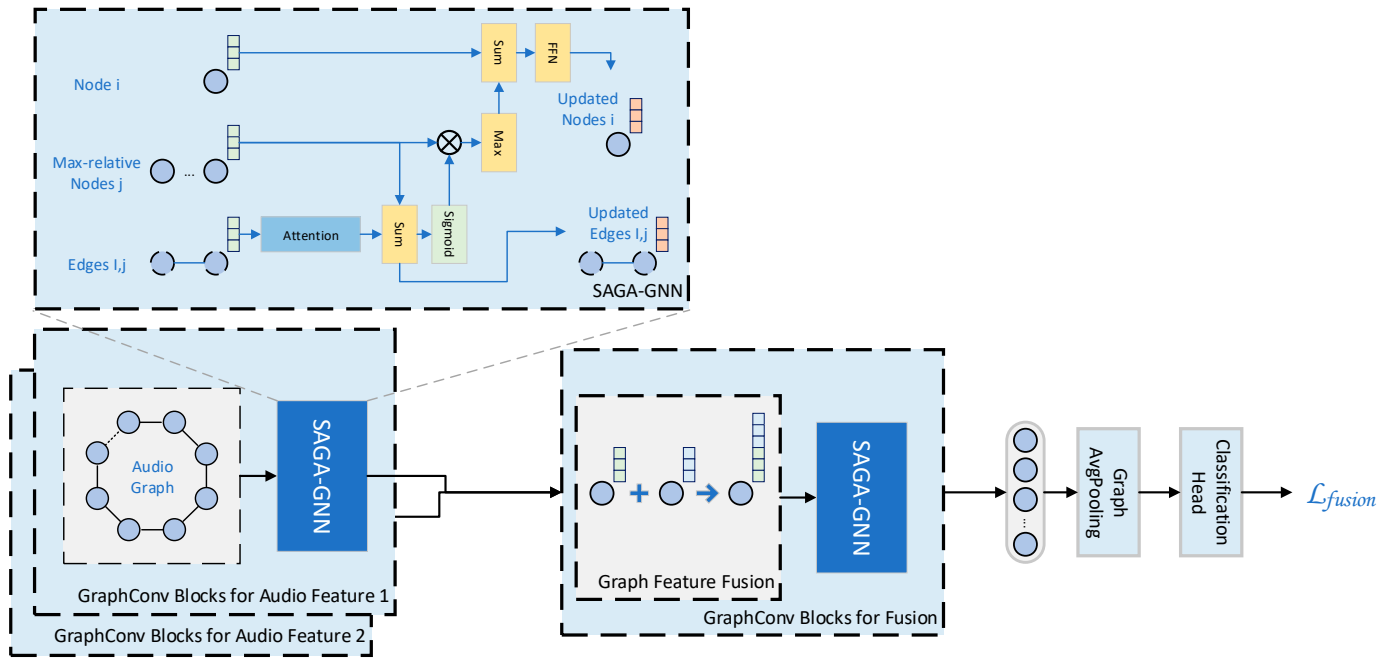


Figure 4. Proposed situation awareness gated-attention GNN for audio features of situation awareness.

Inspired by [40,56], SAGA-GCN instead selects a subset of nodes based on their maximum relevance and employs a gating mechanism [57] to filter out irrelevant nodes through edge feature updating. Algorithm 1 outlines the proposed method for updating the features of nodes and edges.

Algorithm 1: Situation Awareness Gated-Attention GCN.

Input: Graph $G = (V, E)$, node features h^0 , edge features e .

Output: Updated node features h^l

Parameters: Learnable weights W_e^l, U^l, FFN^l

for layer $l = 1$ to L **do**

for each node $v_i \in V$ **do**

 //1. Max-relative Neighbour Selection

 Compute pairwise distances using Equation (17).

 Select max-relative nodes $v_j \in \mathcal{N}_i^k$ using Equation (18).

 //2. Edge Update & Gating Weights

for each $v_j \in \mathcal{N}_i^k$ **do**

 Update edge feature e'_{ij} using Equation (19)

 Update gating weight g_{ij} using Equation (20)

 Update attention-based weights a_{ij} using Equation (21)

 Generate gated-attention weight w_{ij} using Equation (22)

end for

 //3. Gated-Attention Aggregation

$$m_i = \max \left(h_i^l - \left(w_{ij}^l \cdot h_j^l \right) \right)$$

$$h_i^{l+1} = FFN^l(\sigma(m_i)) + h_i^l$$

end for

end for

We organize the SAGA-GCN into three components: (a) max-relative neighbor selection; (b) attention-based edge updating; and (c) gating node aggregation and updating. After obtaining the node features encoded by the convolutional layers and constructing them into a graph $G = (V, E)$, the most relevant node neighbors are selected at each GCN layer. For this purpose, a pairwise distance calculation is performed before every aggregation, which can be represented as follows:

$$d_{ij} = \phi(h_i, h_j) = |h_i - h_j|_2 \quad (17)$$

where $|\cdot|_2$ denotes the Euclidean distance. This distance captures both structural and feature-based dissimilarity.

For each node v_i , select the top- k most relevant neighbors by ranking nodes v_j (where $j \neq i$) based on minimum distance (i.e., maximum relevance):

$$\mathcal{N}_i^{(k)} = \underset{v_j \in V}{\operatorname{argtop}_k}(-d_{ij}) \quad (18)$$

where $\operatorname{argtop}_k(\cdot)$ returns the indices of the k smallest distances. This ensures v_i only connects to the most similar nodes before node aggregation and update.

In graph-based audio processing, as demonstrated by previous work, edge features explicitly encode the relationships between spectral time components (such as harmonic connections between bands or causal event sequences), making pattern discovery more robust compared to implicit feature fusion [38]. Additionally, dynamic edge weights have proven effective in robust audio analysis in the presence of noise [39]. This is crucial for tasks such as anomaly detection in SA. In order to fully utilize the graph composed of audio data, we introduce a gated-attention-based component that dynamically modulates connection strengths according to their relevance to situational awareness events. This approach combines the noise robustness of gating with the contextual awareness of attention, specifically designed for the challenges of SA audio processing.

Given the max-relative neighborhood $\mathcal{N}_i^k$ for node v_i , we define a gated aggregation scheme to dynamically modulate the influence of neighboring nodes $v_j \in \mathcal{N}_i^k$. The edge updating process for updated e'_{ij} begins with feature enhancement, where each connection between node v_i and its selected neighbor $v_j \in \mathcal{N}_i^k$ generates an enriched edge representation, inspired by the following [57]:

$$e'_{ij} = W_e^l \cdot (h_j || e_{ij}) + b_e \quad (19)$$

$$g_{ij} = \operatorname{Sigmoid}(e'_{ij}) \quad (20)$$

where $(\cdot || \cdot)$ denotes the concatenation of vectors. This enhanced edge feature simultaneously encodes the source node characteristics, neighbor node features, and original edge attributes through concatenation followed by learned transformation. The attention weight serves as a relevance scorer, encoding edge features to generate focused feature representations [58]:

$$a_{ij} = e'_{ij} \sigma(\operatorname{Conv1D}_k(\operatorname{GAP}(e'_{ij}))) \quad (21)$$

where $\operatorname{GAP}(\cdot)$ and $\operatorname{Conv1D}_k(\cdot)$ denote the global average pooling layer and 1D convolutional layer with a kernel of k length. These components combine through element-wise multiplication to produce the final gated-attention weight:

$$w_{ij} = a_{ij} g_{ij} \quad (22)$$

With the most relevant node neighbors $\mathcal{N}_i^k$ obtained, the use of max-pooling as an aggregator maintains the generality of node features [56]. The message passing of a SAGA-GCN layer can finally be expressed as follows:

$$h_i^{l+1} = FFN^l \left(\sigma \left(U^l \max \left(h_i^l - \left(w_{ij}^l \cdot h_j^l \right) \right) \right) \right) + h_i^l \quad (23)$$

where $\max(\cdot)$, here, denotes the max-pooling aggregation, U^l denotes the learnable parameters, and $FFN(\cdot)$ denotes a position-wise feed-forward network for further updating and provides additional nonlinear transformations to the model [40]. The second term represents the residual connection.

5. Experiments

In this section, we investigate the effectiveness of different acoustic features on graph structure, followed by analysis of the effectiveness of GNN and attention mechanisms targeting situation awareness audio. The experiments are based on the following research questions: (a) What acoustic features and their combinations are suitable for graph structures? (b) Can graph-based features outperform traditional methods? (c) What is the effect of graph structures on features? (d) What is the effect of the enhancement of graph models with attention mechanisms. In the following subsections, we provide a detailed description of the experimental dataset in our experimental setup and present our results and analysis.

5.1. Experiment Settings

5.1.1. Experiment Datasets

SA is a restricted and complicated subset of general audio. Therefore, we use different SA-related datasets in order to accomplish the tasks of audio anomaly detection as well as disaster-related audio event classification. This subsection describes the datasets we used and the metrics employed in the experiment, as well as their roll with respect to SA.

We adopt the dataset from the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 Task2 [16], which contains normal and abnormal data for six machine types (slide, fan, pump, valve, toy car, and toy conveyor), which can correspond to SA targeting machine status in realistic scenarios. The dataset is partitioned into training and test sets, where the training set contains only normal sound data, while the test set contains both normal and abnormal sound data. Following [16], the models were evaluated on the DCASE dataset using the metrics area under curve (AUC) and partial area under curve (pAUC), with higher AUC reflecting better detection performance of the model in the same category. They can be calculated via the following equations [16]:

$$AUC = \frac{1}{N_- N_+} \sum_{i=1}^{N_-} \sum_{j=1}^{N_+} \mathcal{H} \left(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-) \right), \quad (24)$$

$$pAUC = \frac{1}{|pN_-|N_+} \sum_{i=1}^{|pN_-|} \sum_{j=1}^{N_+} \mathcal{H} \left(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-) \right) \quad (25)$$

with $\mathcal{H}(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$

where $\mathcal{H}(\cdot)$ denotes a hard-threshold function, N_- and N_+ are numbers of normal and abnormal samples, and the sets $\{x_i^+\}_{i=1}^{N_+}$ and $\{x_i^-\}_{i=1}^{N_-}$ are normal and abnormal test samples, respectively. p is set as 0.1 to form a false-positive-rate (FPR) range for the pAUC metric [16].

Additionally, VGGSound [59], an audio–visual dataset proposed by the Visual Geometry Group at University of Oxford is introduced to enhance the representational diversity of SA-related samples to better explore the impact of audio features on models. To reduce training time and more easily explore the effects of features on graph models, we extracted 12 disaster-related classes (i.e., hail, thunder, tornado roaring, volcano explosion, cap gun shooting, machine gun shooting, missile launch, ambulance siren, civil defence siren, fire truck siren, police car siren, smoke detector beeping), which were chosen for their significance in real-world disaster response scenarios from the dataset, for the disaster event classification task, which corresponds to SA for disaster response in realistic scenarios. The extracted dataset contains a total of 9289 video clips, with 50 video clips per class for testing, respectively. We use accuracy as the evaluation metric for the VGGSound dataset, which measures the model’s performance in classifying audio samples into their respective sound categories. It is important to emphasize that despite the original dataset including visual information, our analysis focuses solely on the audio components due to the dataset’s construction based on audio–visual correspondences. This enables us to specifically study the acoustic features’ ability to independently classify disaster events and investigate how graph-based models utilize audio characteristics for SA tasks without being influenced by the potential confounding effects of visual information.

5.1.2. Details of Implementation

We conducted the experiments using the PyTorch 1.11.0 (Meta AI, Menlo Park, CA, USA) framework. To minimize losses, following the hyperparameter settings from [3] and [5], an Adam optimizer [60] with 32 and 128 batches and initial learning rates of 1×10^{-3} and 1×10^{-4} was used for VGGSound and DCASE, respectively. To dynamically adjust the learning rate, we defined a decay rate of 0.1, which was activated when the validation set metrics did not improve within five epochs. Following [5], the frame size of the experimentally implemented audio features is 1024 samples with a 50% overlap, and the number of filter banks is 128.

5.1.3. Audio Features

We employ a variety of commonly used audio features for our experiments. In terms of the feature domain, they can be categorized as follows:

- Time-domain features: i.e., the original waveform (Wav), in which a convolutional layer is used to convert it into a feature shape similar to that of the time–frequency domain for a fair comparison [5,8].
- Spectral domain features: These include the log-Mel spectrogram (Mel) and the gammatone spectrogram (Gam), which use short-time Fourier transform (STFT) to obtain the STFT spectrum, which is subsequently passed through the corresponding filter bank to obtain the frequency selectivity that mimics the human auditory system [61,62]. These spectral domain features have also been shown to be effective in a variety of audio classification tasks [36,63]. Also, the CQT [11] and variable-Q transform (VQT) [64], which design ratios between the center frequencies of adjacent bands in filters to simulate the geometric relationship between pitch frequencies, were originally designed for music signal analysis [11]. This may make them a better representation of audio with overlapping sources or noisy backgrounds, and they were shown to be effective for sound event classification tasks in [65,66].
- Cepstral domain features: These include Mel-frequency cepstrum coefficients (MFCC) and linear frequency cepstrum coefficients (LFCC). They are calculated from the corresponding spectrograms by computing the DCT along the frequency exponent over a fixed time interval and are usually considered as a further abstract representation of

the spectral domain features [67]. Although they were initially designed for speeches, several subsequent studies have demonstrated their effectiveness and applications in other audio classification scenarios, for example, on urban sound classification [13], animal sound recognition [68], and machine condition monitoring [69], etc.

5.2. Exploration of SA-Related Audio Samples

To analyze the frequency composition of SA-related audio samples, we derived frequency activation matrices for different sound categories in the DCASE and VGGSound datasets, following the experimental setup in [63]. These matrices are visualized in Figures 5 and 6, respectively.

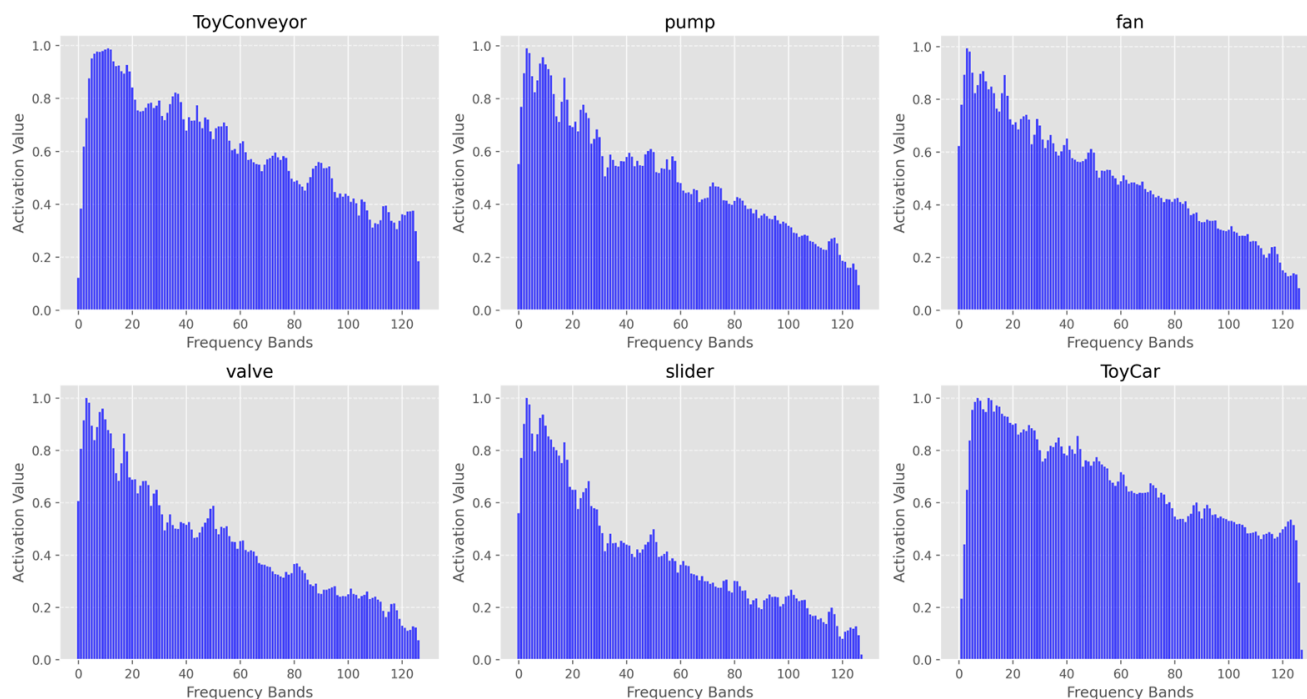


Figure 5. Frequency activation matrices for different sound categories from the DCASE 2020 Task2 dataset.

Given the log-Mel spectrogram $S(t, f)$ of an audio sample, where t represents the time index and f represents the frequency bin (with $f = 1, 2, \dots, 128$ corresponding to the 128 Mel filter banks), we first normalized and averaged all samples of the same category along the time dimension. This process can be expressed as follows:

$$A(f) = \frac{1}{N} \sum_{i=1}^N S_i(t, f) \quad (26)$$

where $A(f)$ is the frequency activation matrix of shape $(128, 1)$, N is the total number of samples in the category, and $S_i(t, f)$ represents the log-Mel spectrogram of the i^{th} sample. Each value in $A(f)$ lies in the range $[0, 1]$, representing the normalized activation level of the corresponding Mel filter bank across all samples. From the resulting matrices, we observed significant variations in activation values across different frequency bands. While the activation patterns vary by category, many categories exhibit higher activation levels in the lower frequency regions. These visualizations reveal several critical dataset characteristics that directly inform our preprocessing pipeline. Specifically, in the DCASE 2020 Task2 dataset, despite all audio categories exhibiting similar deer activation matrices, the toy car category shows higher activation values across various frequency bands, indicating a greater presence of noise in its audio. This phenomenon is even more pronounced in the VGGSound-DR dataset. Additionally, in the VGGSound-DR dataset,

alarm sound categories often exhibit significant activation in specific frequency bands. For instance, the smoke alarm category shows notable activation between the 80th and 100th frequency bands.

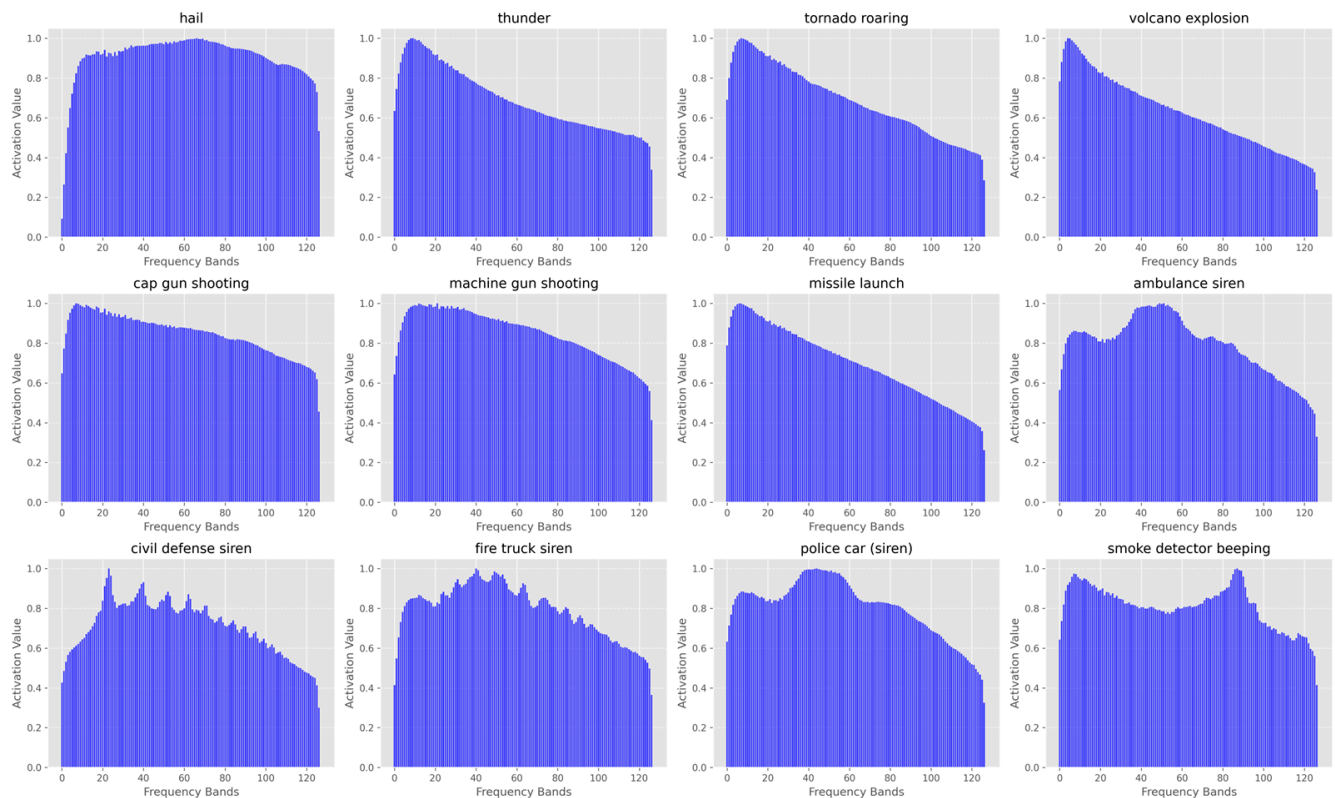


Figure 6. Frequency activation matrices for different sound categories from the VGGSound-DR dataset.

The acoustic feature extraction process employs distinct filter bank configurations to optimize the frequency-domain representations of audio signals through carefully designed spectral resolution trade-offs. As shown in Figure 7, these filter banks comprise arrays of bandpass filters, with each line representing the frequency response curve of an individual filter. The visualization reveals that all filter banks, except the linear-spaced LFCC, implement non-uniform frequency spacing, deliberately allocating more filters to low-frequency regions. This design reflects fundamental psychoacoustic principles, where human hearing demonstrates finer frequency discrimination at lower frequencies, with both the Mel-scale (for Mel spectrograms and MFCCs) and ERB-scale (for gammatone spectrograms) approximating this biological characteristic [61,62]. Comparing the filter densities shows that the Mel filter bank exhibits the most pronounced low-frequency emphasis. In contrast, the linear LFCC bank maintains equal spacing across all frequencies, resulting in comparatively weaker low-frequency resolution. These structural differences directly affect the resulting acoustic features, including low-frequency signal representation (e.g., machinery rumble or vocal fundamentals) and high-frequency content (e.g., crashes or sirens sounds). These frequency-dependent resolution properties significantly influence feature performance in SA tasks.

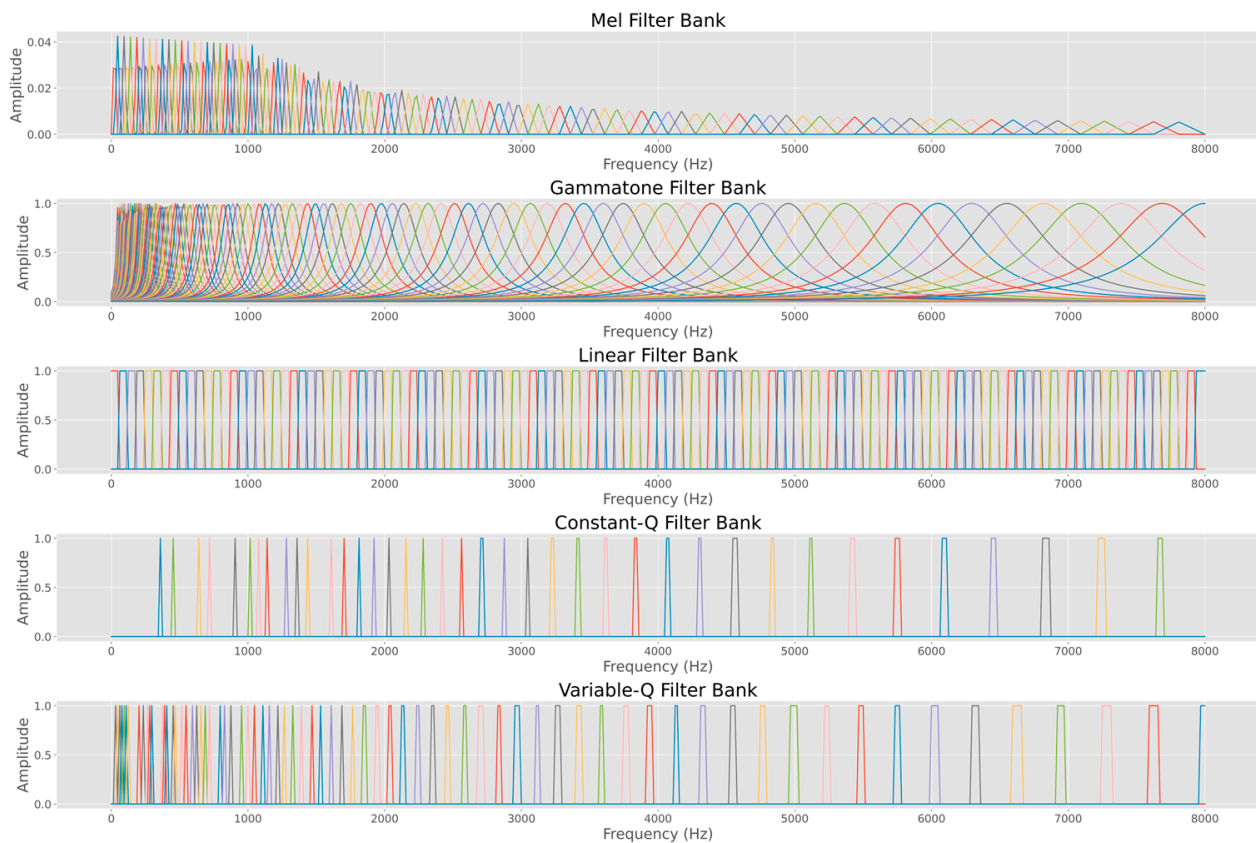


Figure 7. Filter banks of Mel, gammatone, linear, constant-Q, and variable-Q, respectively. Different color lines indicate individual filters.

5.3. Validity of Graph-Based Features for Situation Awareness

Experiments were performed on three graph learning method types, each with four layers (GCN [51], Gated-GCN [57], GAT [29]). In order to compare the performance of the graph learning models, the experiments are set up with four traditional models, including a six-layer CNN [8], EfficientNet-b1 [70], an audio spectrogram transformer (AST) [42], and linear SVM and MLP. All models were trained from scratch to exclude potential advantages due to pre-trained weights. When using SVM and MLP, the acoustic features are flattened to be feedforward. The quantitative results are presented through the corresponding indices of the dataset, as shown in Tables 1 and 2, where the metrics AUC and pAUC for the DCASE 2020 Task2 dataset evaluate the overall discrimination between normal and anomalous sounds, assessing the model's ability to identify deviations from normal operational sounds in industrial equipment monitoring scenarios. Accuracy measures the correct categorization of 12 disaster-related sound classes, validating the model's ability to recognize and distinguish critical acoustic events in emergency situations. Higher values indicate that the model is more reliable and expressive. In addition, t-SNE [71], a technique for the dimensionality reduction and visualization of high-dimensional data, is employed for quantitative results illustration. The distribution of the data can be visualized in a more intuitive way to help explore the appropriate acoustic features and acquire some potential insights.

Table 1. Single-feature performance on the DCASE 2020 Task2 dataset (AUC/pAUC).

Models	Mel	Gam	MFCC	LFCC	CQT	VQT	Wav
GCN	<u>86.6/83.0</u>	80.3/75.8	<i>85.6/82.7</i>	<u>86.4/81.8</u>	75.4/68.7	73.9/68.8	77.1/70.2
GAT	<i>85.7/81.5</i>	80.6/74.9	<u>85.7/82.3</u>	<i>85.9/81.6</i>	75.8/68.8	72.6/68.7	<i>78.6/71.9</i>
Gated-GCN	86.7/83.1	<u>84.9/78.5</u>	87.6/82.7	87.6/81.6	<u>79.3/71.4</u>	<u>76.4/70.2</u>	80.6/73.0
PANN	77.0/71.5	77.4/73.6	76.6/71.4	79.9/73.3	69.5/64.8	69.2/63.6	76.1/71.9
Eff.Net	85.9/81.2	86.3/81.3	85.4/75.8	83.2/76.5	82.8/74.5	82.7/73.7	<u>79.4/71.8</u>
AST	85.7/78.6	<u>85.5/76.0</u>	83.0/76.0	82.7/75.3	76.4/67.8	76.1/68.3	71.8/66.3
MLP	77.5/70.8	<u>72.8/66.9</u>	77.7/71.5	80.2/72.0	63.7/61.8	63.7/61.9	62.4/59.9
SVM	72.9/66.8	62.4/61.1	74.0/68.5	76.3/69.1	63.3/61.8	63.4/61.7	61.6/59.8

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

Table 2. Single-feature performance on the VGGSound-DR dataset (accuracy).

Models	MEL	GAM	MFCC	LFCC	CQT	VQT	Wav
GCN	63.5	58.5	63.7	63.1	63.7	58.0	45.3
GAT	60.6	41.4	63.8	66.5	<i>64.8</i>	61.3	46.2
Gated-GCN	<u>68.0</u>	<u>68.5</u>	69.3	<u>67.0</u>	<u>66.2</u>	<u>63.1</u>	<u>49.7</u>
PANN	65.3	68.2	65.2	65.8	63	56.8	45.6
Eff.Net	72.2	72.2	<u>68.0</u>	67.5	69.7	69	50.8
AST	41.0	42.0	51.6	51.8	42.2	35.2	35.4
SVM	40.9	41.0	43.9	43.7	38.9	35.7	35.2
MLP	38.0	39.4	48.1	48.2	48.7	44.9	38.9

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

5.3.1. Performance of Single Feature

For the quantitative results, Table 1 presents the results of the models for the single-feature case. It is evident that Mel achieves reasonable performance in both models. In the graph model, as well as Mel, MFCC and LFCC also achieve similar performance. Gated-GCN achieved the top performance with the highest AUC (87.6%) and pAUC (83.1%) using Mel and MFCC as input features, respectively, indicating their suitability for both graph models. On the other hand, traditional methods performed well on Gam and LFCC, with the best AUCs of 86.3%, 79.9%, 85.7%, 80.2%, and 76.3% achieved by EfficientNet, PANN, AST, MLP, and SVM, respectively. For the VGGSound-DR dataset, Table 2 shows the results of the graph model and CNN model for the single-feature case. For graph learning methods, Gated-GCN achieved the best accuracy on the MFCC feature (69.3%). For comparison, PANN obtained its best accuracy on gammatone spectrogram (68.2%). However, EfficientNet achieves 72.2%, which is higher than the graph-based models, but it also has a significantly higher number of model parameters. It is worth noting that the standard Transformer-based AST obtains lower results, which fall short of its comprehensible performance on the DCASE dataset. This may be due to the small sample size of SA-related categories, and the lack of inductive bias in the standard Transformer structure leads to its limited generalization effect when trained on fewer data [72].

As is evident from the results in both datasets, as well as the t-SNE visualization discussed further below, there are differences in the acoustic features applied by traditional deep learning models (e.g., CNNs) and graph-based models. Regarding the differences in filter banks, the kernel-based approach of CNNs causes it to treat each frequency interval equally. This aligns with the case of the two better-performing features, i.e., the gammatone spectrogram and LFCC, where the gammatone filter banks have equal response amplitude, and the linear filter banks provide equal frequency-domain resolution. In contrast, the graph-based model focuses more on differentiating each node, whereas Mel and MFCC, which use the Mel filter bank, are based on a higher resolution of the signal at relatively low

frequencies, as well as a larger response amplitude. Additionally, the t-SNE visualization of CNNs also exhibits certain drawbacks compared to graph-based models: the samples have similar representations to each other, appearing as bars in the t-SNE visualization. Furthermore, when the output layer is the same linear layer, the more similar the sample representations are, the more difficult it becomes to fit a reasonable hyperplane for classification, which is reflected in the simulation results.

For the qualitative results, we used t-SNE as a tool to visualize the model and samples to provide some potential insights. For the VGGSound-DR dataset (Figure 8), the t-SNE visualization suggests that the original acoustic features do not clearly distinguish individual audio event samples. However, it can be observed that samples of the same or similar events tend to be represented in closer proximity. For instance, samples from siren categories, which typically exhibit a single frequency and more recognizable patterns, appear more closely grouped. For the DCASE dataset, the visualization indicates that many of the extracted features roughly separate normal and abnormal samples, with varying levels of clarity. Among these, Mel features appear to achieve a greater degree of separation, with distinct sub-clusters within normal and abnormal samples, which is correlating with their superior pAUC score (83.1 achieved by Gated-GCN). This observation aligns with the dataset containing sounds of a diverse range of machines. In contrast, features like Gam show more overlap between normal and abnormal samples. This aligns with its quantitative results (84.9 and 78.5 of AUC and pAUC, respectively, achieved by Gated-GCN), which show a weaker performance compared to similar Mel features. MFCC and LFCC, on the other hand, display the more compact clustering of normal and anomalous samples, respectively. The difference between these two features, due to their reliance on different filter banks (with MFCC emphasizing low frequencies and LFCC treating all frequencies equally), is reflected in the visualizations. MFCC appears to exhibit more distinct cluster boundaries, while LFCC shows the opposite pattern. For this reason, they both achieved the highest AUC scores (87.6). Features like CQT and VQT tend to yield discrete representations for anomalous samples, whereas Wav, a time-domain feature, demonstrates a distribution where normal samples appear more spread out, and abnormal samples are compact but not significantly distant from the normal cluster in the feature space.

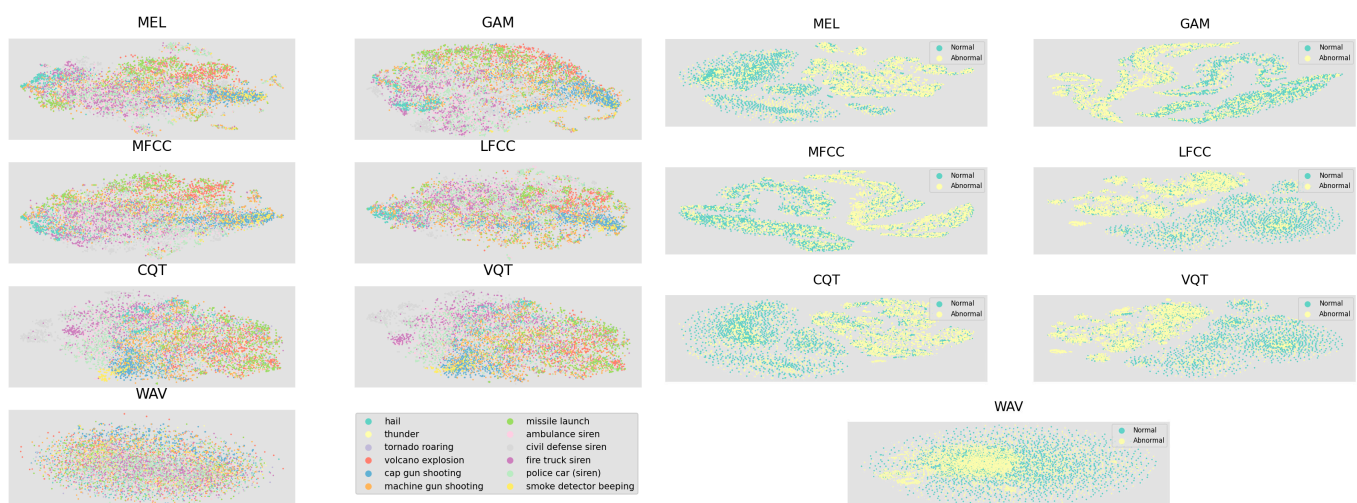


Figure 8. t-SNE visualization for original features from the VGGSound-DR dataset (left) and DCASE 2020 Task2 dataset (right).

For CNN-extracted features, the t-SNE visualizations suggest similar trends across different acoustic features for both datasets (Figure 9). VQT features appear more scattered and exhibit greater category overlaps compared to other features, which may partly explain VQT's lower classification accuracy in the CNN model (Table 1). Conversely, the gammatone spectrogram, which achieves the highest accuracy, does not show clear category demarcations in the visualization, but samples from each category appear more compactly aggregated.

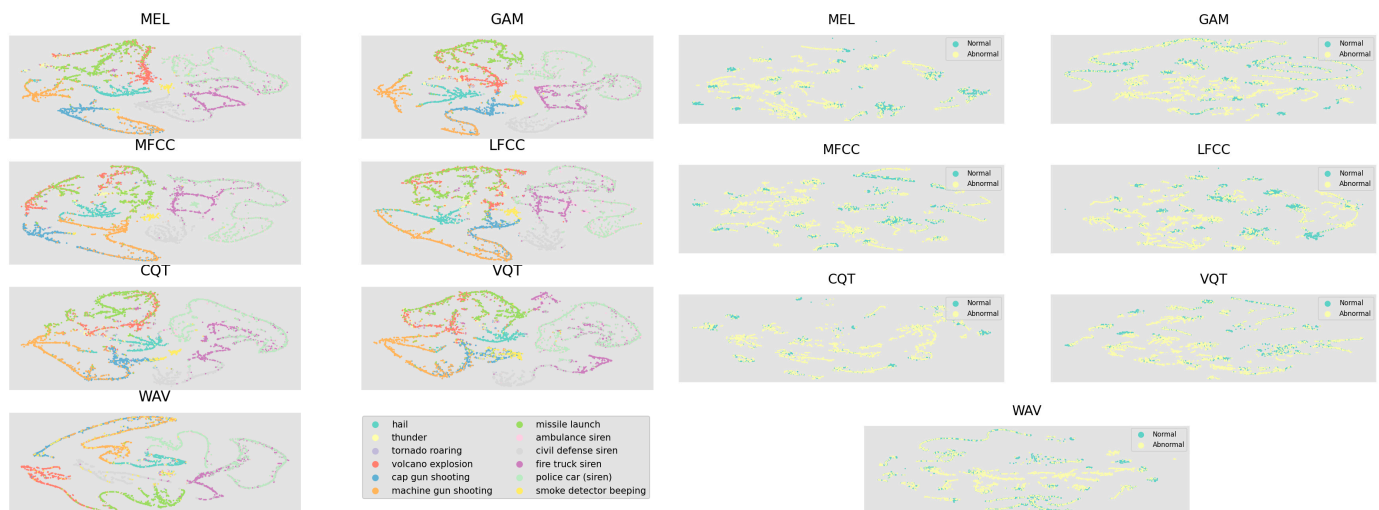


Figure 9. t-SNE visualization for features processed by CNN on the VGGSound-DR dataset (**left**) and DCASE 2020 Task2 dataset (**right**).

The samples from the DCASE dataset are aggregated into multiple dispersed clusters, and each cluster contains both normal and abnormal samples. According to the definition of training, i.e., training with machine category labels instead of normal or abnormal labels, each cluster represents samples from different machine categories. In the figure, cepstral coefficients show the highest separation between normal and abnormal samples. Additionally, the distance between the sample representations is small across the features, leading to difficulty faced by the model in distinguishing whether a sample is normal or not.

In graph learning models, the t-SNE visualizations (Figure 10) suggest that individual categories are more distinctly demarcated when using cepstral coefficients (e.g., MFCC and LFCC) as inputs. However, overlap is observed for categories with similar content, such as siren-related events. In contrast, spectrogram-based features tend to produce more dispersed representations. This supports the observation that cepstral coefficients reduce the impact of noise on audio samples, resulting in slightly lower accuracy for Mel compared to MFCC on the VGGSound-DR dataset (68.0 vs. 69.3). On the other hand, the visualizations suggest that Mel, Gam, MFCC, and LFCC can cluster most of the samples by machine type and degree of normalcy, whereas CQT, VQT, and Wav cluster the samples by machine type, but they still show multiple clusters with evenly distributed normal and abnormal samples, making it difficult for the model to discriminate between them, aligning with the quantitative results.

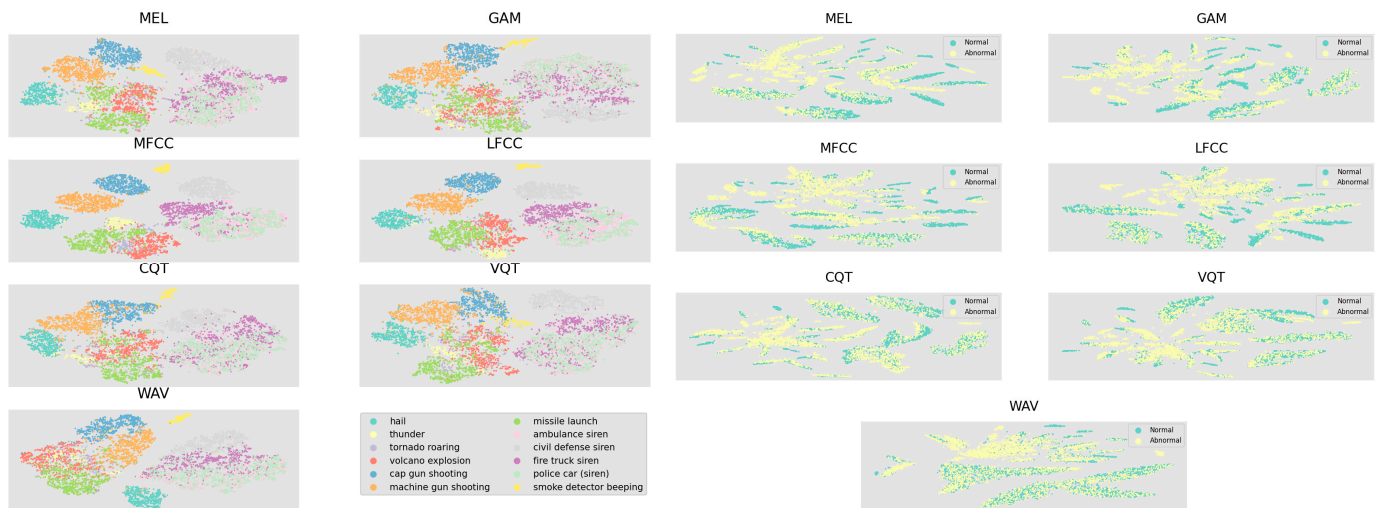


Figure 10. t-SNE visualization for features processed by Gated-GCN on the VGGSound-DR dataset (left) and DCASE 2020 Task2 dataset (right).

5.3.2. Performance of Feature Combinations on Models

Based on the summary of single features in the previous subsection, Mel and MFCC perform better in graph models, while Gam and LFCC perform better on CNNs. Considering the feature extraction methods, the features involved in the experiment can be categorized into several different classes, namely, (1) the spectral domain (Mel, Gam, CQT, VQT), the cepstrum domain (MFCC, LFCC), and the time domain (Wav); and (2) the filter bank differences, i.e., the Mel filter bank (Mel, MFCC), the Gammatone filter bank (Gam), the linear filter bank (LFCC), and the non-applicable filter (Wav). Based on the previous conclusion, CQT, VQT, and Wav can be excluded due to their poor representation in the single-feature case.

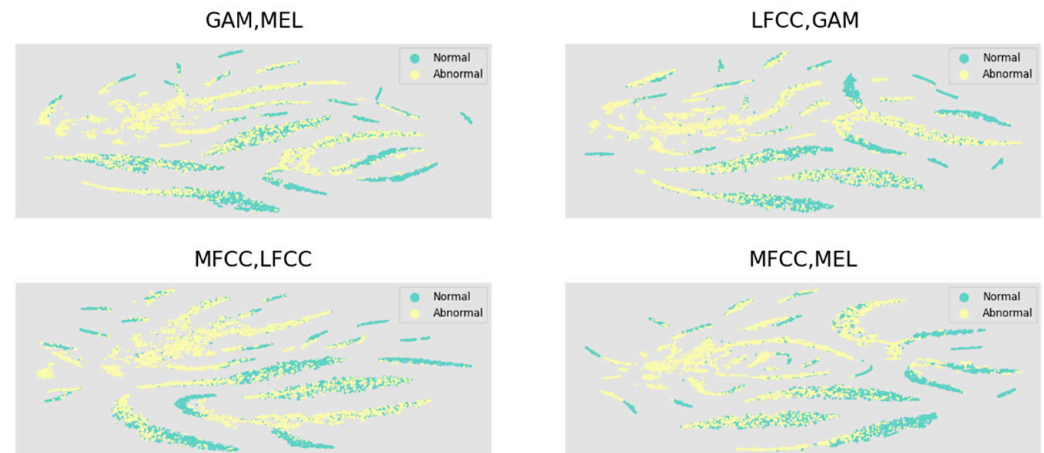
The DCT captures the overall shape of the spectrum by emphasizing the lower-frequency components (which represent the broad structure) and reducing the influence of higher-frequency components (which typically represent finer details). This process effectively condenses the detailed frequency information into a smaller set of coefficients that summarize the signal's global spectral shape, thereby converting spectral-domain features into cepstral-domain features, which are more compact [67]. Additionally, when using cepstral coefficients, it is often necessary to discard higher-dimensional features, i.e., high-frequency information in the spectrum, which makes the cepstral coefficients smoother and less noisy compared to the spectrum.

Based on the possibilities and reasoning outlined above, the experiments were set up in four cases for the graph model and CNN model, namely, Mel and Gam, Mel and MFCC, MFCC and LFCC, and Gam and LFCC, corresponding to different combinations of filter banks and feature domains. Table 3 and Figure 11 illustrate the quantitative results and the feature distributions of the Gated-GCN for these feature combinations after training. The Mel and MFCC combination achieved the highest result, indicating that the same filter bank and different feature domains have positive effects. For CNN, the combination of MFCC and LFCC, but not Gam and LFCC (which produced the two highest results for single features), achieved the highest value.

Table 3. Feature combinations performance (AUC/pAUC).

Models	Mel + Gam	Mel + MFCC	MFCC + LFCC	Gam + LFCC
GCN	87.2/83.3	87.8/83.4	86.9/82.9	87.0/79.5
Gated-GCN	87.5/83.4	88.0/84.2	87.8/84.2	87.5/82.4
PANN	77.9/71.8	77.5/72.6	79.1/70.9	77.8/72.9
EfficientNet	87.0/81.9	86.5/81.4	<u>87.4/75.9</u>	86.8/ <u>81.4</u>

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

**Figure 11.** t-SNE visualization of trained feature combinations for the Gated-GCN.

Comparing the sample distributions of single features trained on the graph model in Figure 10 (for example, the MFCC feature that obtained the highest performance), the addition of Mel makes the distribution of its anomaly samples more compact. Most of the clusters can bring together the overall normal samples, while each separates the normal and abnormal samples (shown in red box), which does not occur in other combinations of features, possibly thanks to the same filter sets. Gated-GCN using feature combinations improves significantly on pAUC and shows an improvement in its performance in recognizing anomalous audio in each machine category.

5.4. Effects of Graph Structure for Situation Awareness Audio Features

Experiments with three graph structures based on Gated-GCN were performed, namely, line graph, cycle graph, and similarity graph. Unlike the fixed graph structure (line and cycle), the similarity graph is more dynamic, and different graphs can be constructed for different audio samples. In contrast to the cycle graph, it can be seen from the adjacency matrix that the similarity graph basically preserves node self-connections, as well as the connections between the front and back nodes. This is because (1) nodes are similar to themselves, with a similarity of 1, and (2) neighboring times in audio always have similar sounds. In Figure 12, the visualization of the Gated-GCN output using t-SNE shows the difference between the different graph structures. The line graph is less distinct compared to the cycle graph, while the similarity graph has clear clusters of aggregated nodes (red boxes). Table 4 shows a comparison of three graph construction methods using Gated-GCN on the DCASE dataset, confirming that the construction method using similarity can effectively improve the performance of the graph model for audio representation.

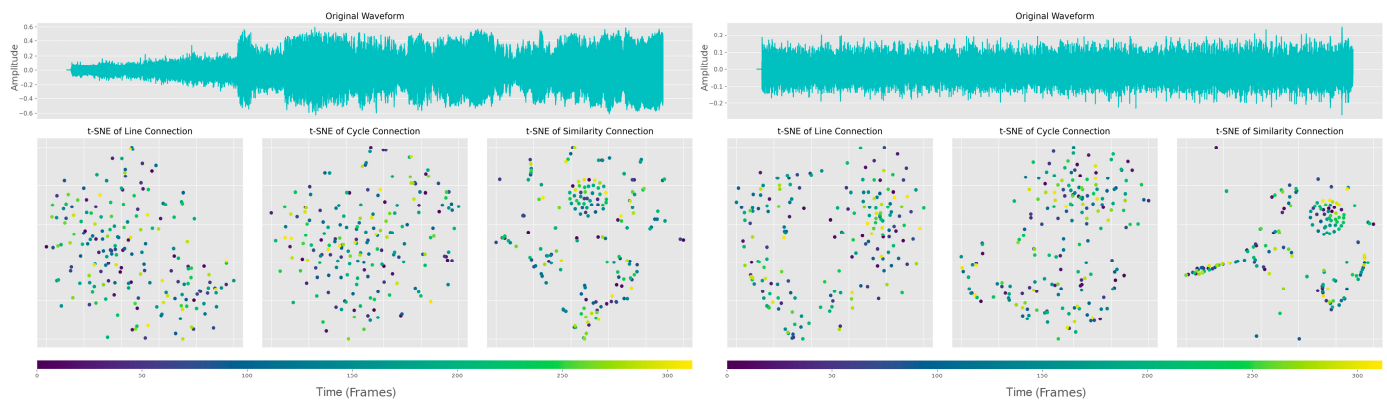


Figure 12. t-SNE visualization of graph nodes output from the learned graph models of different graph construction methods (line, cycle, and similarity).

Table 4. Graph construction methods comparison on the DCASE 2020 Task2 dataset (AUC/pAUC).

Models	Feature	Line	Cycle	Similarity
Gated-GCN	Mel	86.2/82.5	86.7/83.1	87.4/83.5
	MFCC	87.1/81.4	87.6/82.7	88.2/82.9
	Mel + MFCC	87.8/82.9	88.0/84.2	89.4/84.5

The results for the first place are highlighted in **bold**.

5.5. Graph Neural Network with Attention Mechanism

We conducted experiments on the effects of different attention mechanisms on GNN performance using Mel as input, and the results are shown in Table 5. With Gated-GCN serving as the local model, using the self-attention mechanism to provide global attention to the model can effectively improve the model's performance; while using neighbor attention mechanisms further increases the advantage. Based on the combination of the proposed GNN with an attention mechanism and acoustic features suitable for graph modeling, as discussed previously, the performance of the model on the DCASE dataset is shown in Table 6. The proposed model obtains reasonable results on both AUC and pAUC, where the results using Mel and MFCC as input are significantly improved compared to the results of the original Gated-GCN, as shown in Table 3, which demonstrates the effectiveness of using the attention graph as well as introducing global attention. The proposed Gated-GCN with an attention mechanism significantly outperforms the methods in [19,20], which employ CNNs as backbone networks. Furthermore, comparing the scheme of training one model for each machine ID used by the method in [34], the proposed graph-based method is able to achieve higher detection performance using one model. In contrast to the method in [5], which also uses multiple audio features, the proposed method employs a weighted loss function to ensure that each feature is learned in association with the audio labels. It is worth mentioning that ref. [36] uses a generation-based approach, so it requires two stages of training to accomplish the task, i.e., an initial training stage to learn and encode a discrete representation of the samples during the reconstruction of the samples, and a later training stage in which the model is trained for anomaly detection. This means that the proposed method has a more friendly computational cost and storage cost.

Table 5. Performance comparison of GNN with attention mechanisms using Mel spectrogram.

GNN Method	Attention Mechanism	AUC	pAUC
Gated-GCN	None	87.4	83.5
	Self-Attention	88.0	83.8
	Neighborhood Attention	89.2	83.8

The results for the first place are highlighted in **bold**.

Table 6. Performance comparison of models on the DCASE 2020 Task2 dataset.

Methods	Feature	AUC	pAUC
IDNN [19]	Mel Spec.	79.30	63.78
MobileNet-v2 [20]	Mel Spec.	84.00	77.74
Glow Aff [34]	Mel Spec.	85.20	73.90
STgram [5]	Mel Spec. + Wav	89.68	<u>84.64</u>
AudDSR [36]	Mel Spec.	<u>90.12</u>	84.59
Proposed Gated-GCN with Attention	Mel Spec. + MFCC	90.59	84.81

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

5.6. SAGA-GNN

To evaluate the effectiveness of our proposed SAGA-GNN, we conducted extensive comparisons with state-of-the-art audio classification models, including both conventional deep learning approaches and recent graph-based methods (Gated-GCN [57] and Gated-GNN [73]), as well as the SAG-GNN (the version of SAGA-GNN without attention layers). To ensure a fair comparison, we utilized the same feature combination as input for graph learning-based methods.

As shown in Table 7, our SAGA-GNN achieves superior performance compared to existing methods, including IDNN, MobileNet-v2, Glow Aff., STgram, and AudDSR, which rely solely on spectrogram-based representations. Notably, our model outperforms all prior approaches in both AUC (91.26%) and pAUC (85.18%), demonstrating its robustness in detecting anomalous sounds.

Table 7. Performance comparison for SAGA-GNN on the DCASE 2020 Task2 dataset.

Methods	Feature	AUC	pAUC	Param.	Time
IDNN [19]	Mel Spec.	79.30	63.78	\	\
MobileNet-v2 [20]	Mel Spec.	84.00	77.74	\	\
Glow Aff [34]	Mel Spec.	85.20	73.90	\	\
STgram [5]	Mel Spec. + Wav	89.68	<i>84.64</i>	2 M	6 h 12 min
AudDSR [36]	Mel Spec.	<i>90.12</i>	84.59	\	\
Gated-GNN	Mel Spec. + MFCC	86.72	78.28	3 M	7 h 11 min
Gated-GCN	Mel Spec. + MFCC	89.36	84.49	5 M	7 h 45 min
Proposed Gated-GCN with Attention	Mel Spec. + MFCC	<u>90.59</u>	<u>84.81</u>	15 M	9 h 53 min
Proposed SAG-GCN	Mel Spec. + MFCC	91.09	85.05	17 M	8 h 41 min
Proposed SAGA-GNN	Mel Spec. + MFCC	91.26	85.18	18 M	8 h 32 min

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

Furthermore, when compared to other gated graph-based methods (Gated-GCN and Gated-GNN), our model exhibits a clear advantage, highlighting the benefits of max-relative node sampling and adaptive gated attention. While these baseline graph methods already improve upon traditional CNNs, our approach further refines the feature aggregation process by suppressing irrelevant nodes and emphasizing critical acoustic events.

Table 8 presents the classification accuracy on the VGGSound-DR dataset, which evaluates model robustness under domain shifts. Our SAGA-GNN achieves 74.8% accuracy, surpassing not only standard deep learning models (ResNet-50, EfficientNet-b1, PANN) but also other gated graph variants (Gated-GCN, Gated-GNN). This confirms that our method generalizes well across different acoustic conditions, making it particularly suitable for real-world situational awareness applications where environmental noise and domain variations are common.

Table 8. Performance comparison for SAGA-GNN on the VGGSound-DR dataset.

Methods	Feature	Acc.	Param.	Time
ResNet-50	Mel Spec.	65.1	26 M	2 h 20 min
EfficientNet-b1	Mel Spec.	65.8	8 M	2 h 54 min
PANN	Mel Spec.	52.6	75 M	4 h 53 min
Gated-GNN	Mel Spec. + MFCC	66.2	3 M	2 h 18 min
Gated-GCN	Mel Spec. + MFCC	70.7	5 M	2 h 27 min
Proposed Gated-GCN with Attention	Mel Spec. + MFCC	<u>72.4</u>	15 M	2 h 52 min
Proposed SAG-GCN	Mel Spec. + MFCC	74.2	17 M	2 h 45 min
Proposed SAGA-GNN	Mel Spec. + MFCC	74.8	18 M	2 h 42 min

The results for the first, second and third place are highlighted in **bold**, underline and *italics*.

The model parameters and training duration are also presented in Tables 7 and 8, respectively. While the proposed graph learning-based method outperforms baseline models in terms of accuracy, a notable limitation is the higher number of model parameters in the audio anomaly detection task compared to lightweight CNN models. This results in longer training times, with STgram taking 2–3 additional hours. Compared to common CNN models, the proposed model has fewer parameters (lower than Resnet-50 and PANN, which are frequently used in audio classification tasks), but due to the need to restructure audio features into a graph structure, its training time is comparable to that of CNN models. These limitations could lead to longer training or inference times on computing platforms with lower computational power.

The superior performance of SAGA-GNN across both the DCASE 2020 Task2 and VGGSound-DR datasets can be attributed to its dual mechanisms of max-relative neighbor sampling and learnable gated attention, which directly address the unique challenges of SA audio. Unlike conventional graph models that uniformly aggregate node features, SAGA-GNN selectively emphasizes task-relevant audio frames by dynamically constructing graph connections based on feature dissimilarity, thereby minimizing the influence of irrelevant or noisy segments. The gated-attention mechanism further refines this by adaptively modulating edge weights based on contextual importance, enhancing the model's sensitivity to sparse but critical audio events such as alarms, explosions, or anomalies in machine operation. This combination ensures robust representation learning, particularly in the presence of ambient noise or variable event timing, leading to improved anomaly detection and classification accuracy. To visualize, we compared the audio graph node distribution of Gated-GCN and SAGA-GNN on the same SA audio samples (as shown in Figure 13). It is evident that regardless of whether there are fewer event nodes or more

noise nodes, the audio graphs processed by SAGA-GNN are more compact, with fewer scattered nodes. This demonstrates the effectiveness of the proposed method.

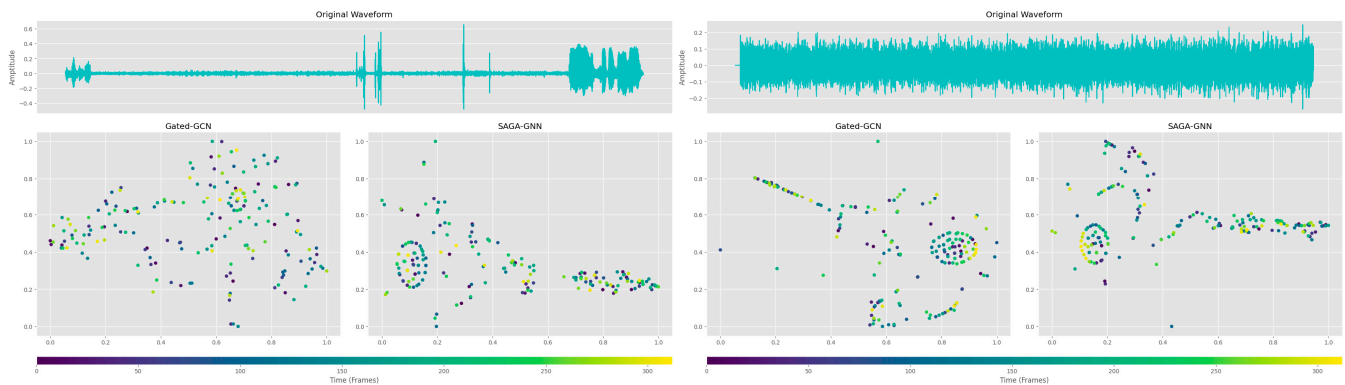


Figure 13. t-SNE visualization of graph nodes output from Gated-GCN and SAGA-GCN.

5.7. Limitations

Despite the demonstrated effectiveness of the proposed SAGA-GNN model in capturing critical audio events for situation awareness tasks, several limitations remain. First, the graph construction method introduces non-negligible computational overhead. Specifically, the use of similarity-based graph structures, which compute pairwise feature similarities and retain top-k neighbors, incurs quadratic complexity with respect to the number of audio frames. While this approach enables richer modeling of relational dependencies than traditional line or cycle graphs, it may be impractical for real-time or resource-constrained applications. Moreover, these similarity graphs—although more expressive than fixed sequential connections—may still fall short of capturing the full spectrum of temporal and semantic dependencies inherent in audio signals. For instance, long-range or multi-scale event correlations may not be fully exploited due to the use of fixed similarity metrics and static top-k thresholds. Future work could explore adaptive or learnable graph construction techniques, potentially informed by external knowledge or hierarchical clustering, to better capture complex relationships among audio segments.

The transparency and trustworthiness of the graph learning framework are another aspect that requires further development [74]. Although the use of attention mechanisms provides some degree of interpretability, by revealing which nodes and edges are weighted more heavily during inference, this interpretability is currently implicit. Given the increasing deployment of machine learning systems in high-stakes domains such as disaster response or industrial monitoring, it is critical to enhance the explainability of graph-based models and establish trust through rigorous evaluation of their behaviors.

6. Conclusions

In conclusion, this study presents a framework integrating graph neural networks with neighborhood attention mechanisms for audio feature representation with application to situation awareness. First, we developed an attention-enhanced graph neural network that improves feature representation through neighborhood attention mechanisms, exploring graph-based features, as well as their combinations. Building upon this foundation, our primary contribution, the situation awareness gated-attention graph neural network (SAGA-GNN), advances the field with two key innovations: (a) max-relevance node sampling method that selectively retains only the most salient acoustic features; (b) adaptive gated attention that dynamically suppresses noise while amplifying critical event patterns. Through comprehensive experimentation, we demonstrated that SAGA-GCN consistently outperforms both traditional audio processing methods and baseline

graph networks. The model's ability to learn robust representations from multi-feature inputs proves particularly valuable in SA applications, where distinguishing between relevant events and environmental noise is paramount. The findings underscore the significance of audio-specific gated-attention mechanisms in graph learning frameworks, as well as the potential of graph-based audio features in situation awareness applications.

Author Contributions: Conceptualization, K.P.S., J.C. and L.M.A.; Methodology, J.C. and K.P.S.; Software, J.C.; Investigation, J.C., K.P.S. and H.X.; Writing—original draft, J.C. and K.P.S.; Writing—review and editing, K.P.S. and L.M.A.; Supervision, K.P.S. and J.S.; Project administration, K.P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study is publicly available in [16].

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Endsley, M.R. Automation and Situation Awareness. In *Automation and Human Performance: Theory and Applications*; Parasuraman, R., Mouloua, M., Eds.; Lawrence Erlbaum: Mahwah, NJ, USA, 1996; pp. 163–181.
- Sui, L.; Guan, X.; Cui, C.; Jiang, H.; Pan, H.; Ohtsuki, T. Graph Learning Empowered Situation Awareness in Internet of Energy With Graph Digital Twin. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7268–7277. [\[CrossRef\]](#)
- Chen, J.; Seng, K.P.; Ang, L.M.; Smith, J.; Xu, H. AI-Empowered Multimodal Hierarchical Graph-Based Learning for Situation Awareness on Enhancing Disaster Responses. *Future Internet* **2024**, *16*, 161. [\[CrossRef\]](#)
- Zhao, D.; Ji, G.; Zhang, Y.; Han, X.; Zeng, S. A Network Security Situation Prediction Method Based on SSA-GResNeSt. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 3498–3510. [\[CrossRef\]](#)
- Liu, Y.; Guan, J.; Zhu, Q.; Wang, W. Anomalous Sound Detection Using Spectral-Temporal Information Fusion. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 816–820.
- Mnasri, Z.; Rovetta, S.; Masulli, F. Anomalous Sound Event Detection: A Survey of Machine Learning Based Methods and Applications. *Multimed. Tools Appl.* **2022**, *81*, 5537–5586. [\[CrossRef\]](#)
- Mistry, Y.D.; Birajdar, G.K.; Khodke, A.M. Time-Frequency Visual Representation and Texture Features for Audio Applications: A Comprehensive Review, Recent Trends, and Challenges. *Multimed. Tools Appl.* **2023**, *82*, 36143–36177. [\[CrossRef\]](#)
- Kong, Q.; Cao, Y.; Iqbal, T.; Wang, Y.; Wang, W.; Plumbley, M.D. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2880–2894. [\[CrossRef\]](#)
- Chen, J.; Seng, K.P.; Smith, J.; Ang, L.-M. Situation Awareness in AI-Based Technologies and Multimodal Systems: Architectures, Challenges and Applications. *IEEE Access* **2024**, *12*, 88779–88818. [\[CrossRef\]](#)
- Davis, S.; Mermelstein, P. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [\[CrossRef\]](#)
- Brown, J.C. Calculation of a Constant Q Spectral Transform. *J. Acoust. Soc. Am.* **1991**, *89*, 425–434. [\[CrossRef\]](#)
- Ruinskiy, D.; Lavner, Y. An Effective Algorithm for Automatic Detection and Exact Demarcation of Breath Sounds in Speech and Song Signals. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 838–850. [\[CrossRef\]](#)
- Piczak, K.J. ESC: Dataset for Environmental Sound Classification. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1015–1018.
- Birajdar, G.K.; Patil, M.D. Speech and Music Classification Using Spectrogram Based Statistical Descriptors and Extreme Learning Machine. *Multimed. Tools Appl.* **2019**, *78*, 15141–15168. [\[CrossRef\]](#)
- Breebaart, J.; McKinney, M.F. Features for Audio Classification. In *Algorithms in Ambient Intelligence*; Verhaegh, W.F.J., Aarts, E., Korst, J., Eds.; Springer: Dordrecht, The Netherlands, 2004; pp. 113–129; ISBN 978-94-017-0703-9.
- Koizumi, Y.; Kawaguchi, Y.; Imoto, K.; Nakamura, T.; Nikaido, Y.; Tanabe, R.; Purohit, H.; Suefusa, K.; Endo, T.; Yasuda, M.; et al. Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring. *arXiv* **2020**. [\[CrossRef\]](#)
- Shashidhar, R.; Patilkulkarni, S.; Puneeth, S.B. Combining Audio and Visual Speech Recognition Using LSTM and Deep Convolutional Neural Network. *Int. J. Inf. Technol.* **2022**, *14*, 3425–3436. [\[CrossRef\]](#)
- Nanni, L.; Costa, Y.M.G.; Lucio, D.R.; Silla, C.N.; Brahnam, S. Combining Visual and Acoustic Features for Audio Classification Tasks. *Pattern Recognit. Lett.* **2017**, *88*, 49–56. [\[CrossRef\]](#)

19. Suefusa, K.; Nishida, T.; Purohit, H.; Tanabe, R.; Endo, T.; Kawaguchi, Y. Anomalous Sound Detection Based on Interpolation Deep Neural Network. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 271–275.
20. Wilkinghoff, K. Self-Supervised Learning for Anomalous Sound Detection. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 276–280.
21. Han, K.; Wang, Y.; Guo, J.; Tang, Y.; Wu, E. Vision GNN: An Image Is Worth Graph of Nodes. *arXiv* **2022**, arXiv:2206.00272.
22. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 7370–7377. [\[CrossRef\]](#)
23. Xu, H.; Seng, K.P.; Ang, L.-M. New Hybrid Graph Convolution Neural Network with Applications in Game Strategy. *Electronics* **2023**, *12*, 4020. [\[CrossRef\]](#)
24. Singh, S.; Benetos, E.; Phan, H.; Stowell, D. LHGNN: Local-Higher Order Graph Neural Networks For Audio Classification and Tagging. *arXiv* **2025**. [\[CrossRef\]](#)
25. Bhattacharjee, A.; Singh, S.; Benetos, E. GraFPrint: A GNN-Based Approach for Audio Identification. *arXiv* **2024**. [\[CrossRef\]](#)
26. Castro-Ospina, A.E.; Solarte-Sanchez, M.A.; Vega-Escobar, L.S.; Isaza, C.; Martínez-Vargas, J.D. Graph-Based Audio Classification Using Pre-Trained Models and Graph Neural Networks. *Sensors* **2024**, *24*, 2106. [\[CrossRef\]](#)
27. Aironi, C.; Cornell, S.; Principi, E.; Squartini, S. Graph-based Representation of Audio signals for Sound Event Classification. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; pp. 566–570.
28. Kim, J.-W.; Yoon, C.; Jung, H.-Y. A Military Audio Dataset for Situational Awareness and Surveillance. *Sci. Data* **2024**, *11*, 668. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903.
30. Rampášek, L.; Galkin, M.; Dwivedi, V.P.; Luu, A.T.; Wolf, G.; Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Curran Associates Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 14501–14515.
31. Dwivedi, V.P.; Bresson, X. A Generalization of Transformer Networks to Graphs. *arXiv* **2021**, arXiv:2012.09699.
32. Munir, A.; Aved, A.; Blasch, E. Situational Awareness: Techniques, Challenges, and Prospects. *AI* **2022**, *3*, 55–77. [\[CrossRef\]](#)
33. Lamsal, R.; Harwood, A.; Read, M.R. Socially Enhanced Situation Awareness from Microblogs Using Artificial Intelligence: A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–38. [\[CrossRef\]](#)
34. Dohi, K.; Endo, T.; Purohit, H.; Tanabe, R.; Kawaguchi, Y. Flow-Based Self-Supervised Density Estimation for Anomalous Sound Detection. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 336–340.
35. Ekpezu, A.O.; Wiafe, I.; Katsriku, F.; Yaokumah, W. Using Deep Learning for Acoustic Event Classification: The Case of Natural Disasters. *J. Acoust. Soc. Am.* **2021**, *149*, 2926–2935. [\[CrossRef\]](#)
36. Zavrtanik, V.; Marolt, M.; Kristan, M.; Skočaj, D. Anomalous Sound Detection by Feature-Level Anomaly Simulation. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 1466–1470.
37. Shirian, A.; Guha, T. Compact Graph Architecture for Speech Emotion Recognition. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 6284–6288.
38. Hou, Y.; Song, S.; Yu, C.; Wang, W.; Botteldooren, D. Audio Event-Relational Graph Representation Learning for Acoustic Scene Classification. *IEEE Signal Process. Lett.* **2023**, *30*, 1382–1386. [\[CrossRef\]](#)
39. Gao, Y.; Zhao, H.; Zhang, Z. Adaptive Speech Emotion Representation Learning Based On Dynamic Graph. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 1116–1120.
40. Singh, S.; Steinmetz, C.J.; Benetos, E.; Phan, H.; Stowell, D. ATGNN: Audio Tagging Graph Neural Network. *IEEE Signal Process. Lett.* **2024**, *31*, 825–829. [\[CrossRef\]](#)
41. Shirian, A.; Tripathi, S.; Guha, T. Dynamic Emotion Modeling With Learnable Graphs and Graph Inception Network. *IEEE Trans. Multimed.* **2022**, *24*, 780–790. [\[CrossRef\]](#)
42. Gong, Y.; Chung, Y.-A.; Glass, J. AST: Audio Spectrogram Transformer. In Proceedings of the Interspeech 2021, Brno, Czechia, 30 August–3 September 2021; pp. 571–575. [\[CrossRef\]](#)
43. Xiao, F.; Guan, J.; Zhu, Q.; Wang, W. Graph Attention for Automated Audio Captioning. *IEEE Signal Process. Lett.* **2023**, *30*, 413–417. [\[CrossRef\]](#)

44. Gao, Y.; Zhao, H.; Xiao, Y.; Zhang, Z. GCFormer: A Graph Convolutional Transformer for Speech Emotion Recognition. In Proceedings of the International Conference on Multimodal Interaction, Paris, France, 9–13 October 2023; ACM: New York, NY, USA, 2023; pp. 307–313.
45. Rampášek, L.; Galkin, M.; Dwivedi, V.P.; Luu, A.T.; Wolf, G.; Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv* **2023**. [CrossRef]
46. Sun, C.; Jiang, M.; Gao, L.; Xin, Y.; Dong, Y. A novel Study for Depression Detecting Using Audio Signals Based on Graph Neural Network. *Biomed. Signal Process. Control* **2024**, *88*, 105675. [CrossRef]
47. Ma, X.; Sun, P.; Gong, M. An Integrative Framework of Heterogeneous Genomic Data for Cancer Dynamic Modules Based on Matrix Decomposition. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 305–316. [CrossRef]
48. Bresson, X.; Laurent, T. Residual Gated Graph ConvNets. *arXiv* **2018**, arXiv:1711.07553.
49. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
50. Hassani, A.; Walton, S.; Li, J.; Li, S.; Shi, H. Neighborhood Attention Transformer. *arXiv* **2023**. [CrossRef]
51. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**. [CrossRef]
52. Swietojanski, P.; Braun, S.; Can, D.; Da Silva, T.F.; Ghoshal, A.; Hori, T.; Hsiao, R.; Mason, H.; McDermott, E.; Silovsky, H.; et al. Variable Attention Masking for Configurable Transformer Transducer Speech Recognition. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
53. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv* **2023**. [CrossRef]
54. Hou, Y.; Ren, Q.; Song, S.; Song, Y.; Wang, W.; Botteldooren, D. Multi-Level Graph Learning For Audio Event Classification And Human-Perceived Annoyance Rating Prediction. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 716–720.
55. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 978-0-262-33737-3.
56. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. DeepGCNs: Can GCNs Go As Deep As CNNs? In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 20–26 October 2019; pp. 9267–9276.
57. Dwivedi, V.P.; Joshi, C.K.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks. *J. Mach. Learn. Res.* **2023**, *24*, 1–48.
58. Noman, A.; Beiji, Z.; Zhu, C.; Alhabib, M.; Al-sabri, R. FEGGNN: Feature-Enhanced Gated Graph Neural Network for Robust Few-Shot Skin Disease Classification. *Comput. Biol. Med.* **2025**, *189*, 109902. [CrossRef]
59. Chen, H.; Xie, W.; Vedaldi, A.; Zisserman, A. Vggsound: A Large-Scale Audio-Visual Dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 721–725.
60. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**. [CrossRef]
61. Vorländer, M. Acoustic Measurements. In *Handbook of Engineering Acoustics*; Müller, G., Möser, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 23–52. ISBN 978-3-540-69460-1.
62. Slaney, M. An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank. 1997. Available online: <https://engineering.purdue.edu/~malcolm/apple/tr35/PattersonsEar.pdf> (accessed on 19 June 2025).
63. Mu, W.; Yin, B.; Huang, X.; Xu, J.; Du, Z. Environmental Sound Classification Using Temporal-Frequency Attention Based Convolutional Neural Network. *Sci. Rep.* **2021**, *11*, 21552. [CrossRef] [PubMed]
64. Schörkhuber, C.; Klapuri, A.; Holighaus, N.; Dörfler, M. A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution. In *Semantic Audio*; Audio Engineering Society: New York, NY, USA, 2014.
65. Lidy, T.; Schindler, A. CQT-Based Convolutional Neural Networks for Audio Scene Classification. 2016. Available online: https://dcase.community/documents/challenge2016/technical_reports/DCASE2016_Lidy_4007.pdf (accessed on 19 June 2025).
66. Venkatesh, S.; Koolagudi, S.G. Polyphonic Sound Event Detection Using Modified Recurrent Temporal Pyramid Neural Network. In *Computer Vision and Image Processing*; Kaur, H., Jakhetiya, V., Goyal, P., Khanna, P., Raman, B., Kumar, S., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 554–564.
67. Anden, J.; Mallat, S. Deep Scattering Spectrum. *IEEE Trans. Signal Process.* **2014**, *62*, 4114–4128. [CrossRef]
68. Nolasco, I.; Singh, S.; Morfi, V.; Lostonlen, V.; Strandburg-Peshkin, A.; Vidaña-Vila, E.; Gill, L.; Pamula, H.; Whitehead, H.; Kiskin, I.; et al. Learning to Detect an Animal Sound from Five Examples. *Ecol. Inform.* **2023**, *77*, 102258. [CrossRef]
69. Pichler, C.; Neumayer, M.; Schweighofer, B.; Feilmayr, C.; Schuster, S.; Wegleiter, H. Acoustic-Based Detection Technique for Identifying Worn-Out Components in Large-Scale Industrial Machinery. *IEEE Sens. Lett.* **2023**, *7*, 1–4. [CrossRef]
70. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
71. van der Maaten, L.; Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

72. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
73. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated Graph Sequence Neural Networks. *arXiv* **2017**. [[CrossRef](#)]
74. Zhang, H.; Wu, B.; Yuan, X.; Pan, S.; Tong, H.; Pei, J. Trustworthy Graph Neural Networks: Aspects, Methods, and Trends. *Proc. IEEE* **2024**, *112*, 97–139. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.