GIFF: Graph Iterative Attention Based Feature Fusion for Collaborative Perception

Ahmed N. Ahmed[a], Siegfried Mercelis[b] and Ali Anwar[c]
Imec Research Group, IDLab, Faculty of Applied Engineering, University of Antwerp, 2000 Antwerp, Belgium
{ahmed.ahmed, siegfried.mercelis, ali.anwar}@uantwerpen.be

Keywords: Collaborative Perception, Autonomous Driving, Attention, Graphs, Object Detection.

Abstract: Multi-agent collaborative perception has gained significant attention due to its ability to overcome the challenges stemming from the limited line-of-sight visibility of individual agents that raised safety concerns for autonomous navigation. This paper introduces GIFF, a graph-based iterative attention collaborative perception framework designed to improve situational awareness among multi-agent systems, including vehicles and roadside units. GIFF enhances autonomous driving perception by fusing perceptual data shared among neighboring agents, allowing agents to “see” through occlusions, detect distant objects, and increase resilience to sensor noise and failures, at low computational cost. To achieve this, we propose a novel framework that integrates both channel and spatial attention mechanisms, learned iteratively and in parallel. We evaluate our approach on object detection task using the V2X-Sim and OPV2V datasets by conducting extensive experiments. GIFF has demonstrated effectiveness compared to state-of-the-art methods and has proved to achieve notable improvements in average precision and the number of model parameters.

1 INTRODUCTION

Situational awareness is an important topic in the field of autonomous driving. Autonomous vehicles (AV) mainly rely on onboard sensors to perceive their surrounding environment. However, as shown in Fig. 1, the onboard sensors deployed on the AV are limited by the sensor’s field of view, and horizontal range, due to that the perception system becomes susceptible to many challenges such as occlusion and long-distance perception sparsity, which hinder the situational awareness ability of the AV. While deep learning has improved the perception stacks with data-driven techniques (Qian et al., 2022), the perception module in AV to date is still brittle, especially in the face of extreme situations and corner cases that can lead to catastrophic scenarios. In recent years, there has been an increasing amount of research focused on collaborative perception enabling the vehicle to communicate with neighboring AVs and roadside units to achieve Vehicle-to-Everything (V2X) (Ahmed et al., 2024a; Ahmed et al., 2024b; Li et al., 2021; Wang et al., 2020) significantly improving the situational awareness abilities, a simplified illustration is shown in Fig. 1. With the advent of telecommunication technology developments, collaborative perception (Han et al., 2023) is becoming a promising paradigm that enables sensor information to be shared between neighboring agents (for simplicity, we refer to vehicles and roadside units as agents) in real time. The collaborative perception module operates by intelligently aggregating visual data from multiple relevant agents within the communication range to enhance visual reasoning and detection precision as shown in Fig. 1. In practice, the efficacy of collaborative perception hinges on what data to transmit within the limited network bandwidth and how to aggregate the information received from other agents to build a coherent situational awareness of the surroundings. Due to the topological nature of this problem, in this work, we propose a graph iterative attention-based network to aggregate the ego agent’s local observations with those of neighboring agents. By utilizing both the attention mechanism to attend only to the relevant region of the information provided by the neighboring agents and message-passing functionality within the graph networks, our methodology yields in enhanced situational awareness.

On the one hand, various types of graph neural networks have been proposed (Zhou et al., 2020; Wu et al., 2020), and have proved to be effective for fea-

[a] https://orcid.org/0000-0002-7192-699X
[b] https://orcid.org/0000-0001-9355-6566
[c] https://orcid.org/0000-0002-5523-0634

Ahmed, A. N., Mercelis, S. and Anwar, A.
GIFF: Graph Iterative Attention Based Feature Fusion for Collaborative Perception.
DOI: 10.5220/0013297900003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 820-829
ISBN: 978-989-758-728-3; ISSN: 2184-4321

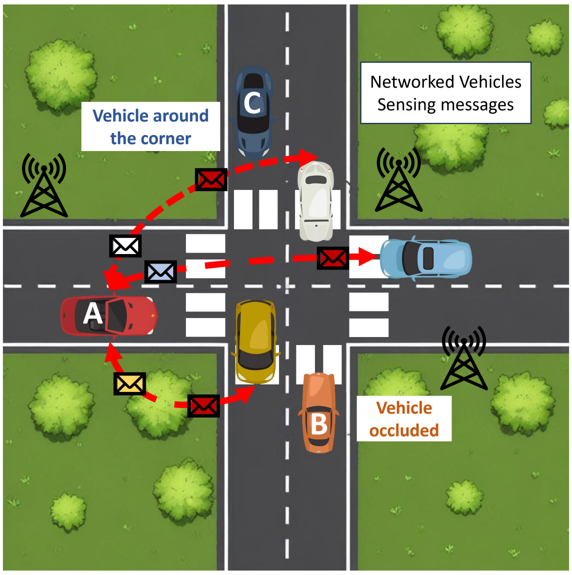
Vehicle around the corner
C
Networked Vehicles Sensing messages
A
B
Vehicle occluded

Figure 1: Illustration of single-agent perception challenges. From the perspective of the ego vehicle (A), vehicle (B) is occluded by the yellow vehicle. Likewise, Vehicle (C) is around the corner lying outside the perception range of Vehicle (A). These typical road scenarios cause dangerous collision risks. If vehicles are given the ability to inform each other “what they can see” achieving collaborative perception those collisions will be avoided.

ture aggregation (Ahmed et al., 2021). We chose GATs (Veličković et al., 2017) to be our core aggregation method, as it exploits the underlying graph structure of the multi-agent collaborative perception data aggregation problem by utilizing the message passing among nodes and attention in the graph. The node features are updated by aggregating node features from the neighbors. Addressing the collaborative perception problem through a graph-based approach allows for the embedding of both ego and received feature maps as graph nodes. This method enables the model to learn edge weights and attention coefficients, which adaptively weigh nodes and their associated features based on inter-node correlations. On the other hand, the benefit of incorporating attention within our proposed feature aggregation scheme is that attention enhances the representation power by directing the model’s to focus on the significant regions within the fused semantic information and suppressing unnecessary ones. In this work, we utilize channel and spatial attention modules to attend to both local and global contexts. We also introduce an iterative attention fusion approach to further refine the feature fusion process, further improving the quality of the final fused feature. The contributions of this work can be summarized as follows:

- Our method proposes a novel methodology for aggregation of informative features on channel-spatial dimensions and incorporating it within the GATs method which simultaneously aggregates complementary information from connected nodes
- Our proposed attention learning network is designed so that channel and spatial attention are learned separately, allowing the model to analyze spatial and channel information without the bias introduced by the correlations between channel and spatial features.
- We propose an iterative attention learning strategy that gradually builds up a richer, more nuanced understanding of the fused features progressively down-weighting less relevant information and focusing on the most significant elements. This further improves the model performance with a substantial increase in the model’s learnable parameters
- We validate our work using a large open dataset V2XSim (Li et al., 2022) which includes LiDAR data retrieved from both vehicles and roadside units. We also perform an extensive ablation study to investigate the performance gain of our proposed design choices.

The rest of this paper is organized as follows. Section. 2 introduces the related work published in recent years. Section 3 describes our proposed method in detail. The experimental results are given in section. 4 and 5, then we perform an ablation study in section. 6 we conclude the paper in section. 7.

2 RELATED WORKS

Graphs have been extensively applied in collaborative perception due to their capability to propagate and aggregate information across neighboring nodes, effectively updating each node’s feature representation. The importance of attention mechanisms in enhancing computer vision tasks has also been well established in prior literature (Guo et al., 2022). Consequently, numerous studies have explored combining graphs with attention mechanisms to improve information aggregation among collaborating agents. The authors in (Zhou et al., 2022) implemented GNN in multi-robot systems by modeling each robot as a graph node and leveraging message-passing combined with cross-attention encoding to enable information sharing and fusion within the team. In the domain of AVs, V2VNet (Wang et al., 2020) employed GNNs to aggregate shared neural features for joint detection and prediction; however, this approach used a convolutional gated recurrent unit for message aggregation, which significantly increased model parameters. DiscoNet (Li et al., 2021) introduced a teacher-student framework that applied a matrix-valued edge weight within the graph to learn node interactions. V2X-ViT (Xu et al., 2022a) examined the use of attention alone by utilizing a vision transformer with window attention for V2X collaboration, though it requires the transmission of full feature maps, increasing bandwidth usage. In addition, Where2comm (Hu et al., 2022) utilized attention on ego and received feature maps to assess correlations among agents. Despite its advantages, Where2comm lacks flexibility in allowing ego agents to adjust their perceptual focus based on immediate environmental conditions, potentially reducing effectiveness in limited communication scenarios. CollabGAT (Ahmed et al., 2024a) incorporated spatial and channel attention in a sequential setup, following the CBAM (Woo et al., 2018) model; however, this approach may not fully capture complex interdependencies between channel and spatial features. Alternatively, the authors in (Ahmed et al., 2024b) integrated spatial and channel attention in a parallel arrangement within their collaborative graph, differing from the sequential arrangement in CollabGAT (Ahmed et al., 2024a). In contrast, our proposed method proposes a graph-iterative attention-based method that incorporates both channel and spatial attention in an iterative manner that learns interdependent patterns in both dimensions parallelly.

3 METHODOLOGY

3.1 Overview

The goal of our proposed method is to aggregate information received from other agents to help enhance the ego agent's situational awareness. In our proposed collaborative perception scheme, we assume that the environment consists of N agents equipped with LiDARs, and their point cloud observations $\mathcal{X} = \{\mathcal{X}_i, \mathcal{X}_j, .., \mathcal{X}_N\}$. In principle, agents can transmit all their retrieved raw point cloud data to the ego agent to aggregate them. However, in practice, we have to consider the network bandwidth limit, as sharing the raw point cloud data among neighboring agents can overload the network, causing huge transmission delays. Thus, we derive a distributed and efficient information-fusing framework that is able to: (i) maximize the object detection accuracy, for the ego agent and (ii) minimize the size of the shared data to prevent bandwidth overloading. The overview of our methodology is presented in Fig. 2.

In that regard, the raw point cloud of each agent $\mathcal{X}_N$ is processed using a unified feature extractor (section. 3.2) into compact semantic representation, named feature map F_N to be transmitted through the V2X channels in real-time. Subsequently, these features are fed into a compression block, further compressing this feature map to further reduce its size to prepare for transmission(section. 3.3). Afterward, using the compressed feature map and the pose of the broadcasting agent we create a collaborative perception message (CPM) to be broadcast to neighboring agents. The ego (receiving) agent decompresses the CPM and passes it to the collaborator selection module (section. 3.4) which selects only relevant agents based on pre-defined metrics (section. 3.4). Eventually, the feature map of the selected agents transformed to the ego agent perspective (section. 3.4). The ego agent and the transformed features are then fed into the feature fusion network to iteratively aggregate all the received feature maps taking into account the relevancy of the neighboring agents to the ego agent (section. 3.5). The fused features are then forwarded to the decoder network (section 3.6) to generate predictions on the final outputs in object detection.

3.2 Feature Extractor

To alleviate communication overhead, each agent independently processes its own LiDAR data, encoding raw point clouds into semantic information, as illustrated in Fig. 2. Specifically, each agent transforms its collected point cloud data, $\mathcal{X}$, into a bird's-eye-view (BEV) representation, flattened along the height dimension. This BEV representation is then inputted into a feature extractor, denoted by $\Theta(\cdot)$, to produce a feature map, $F_i = \Theta(BEV_i)$, where $F_i \in \mathbb{R}^{W \times H \times C}$, with W, H, and C representing the width, height, and channel dimensions of the feature map, respectively. Our approach assumes homogeneous intermediate collaborative perception; thus, all agents utilize the same feature extractor architecture, sharing the same $\Theta(\cdot)$. The primary objective of this work is to improve the effectiveness of the feature map fusion strategy and to evaluate our proposed aggregation approach against state-of-the-art models. To achieve this, we benchmark our intermediate feature aggregation methods by employing the feature extractors from DicoNet (Li et al., 2021) and V2VNet (Wang et al., 2020); enabling an independent analysis of the proposed fusion strategies across different feature extractor architectures. Further details of this analysis are provided in Section 6.

3.3 Compression and Sharing

To minimize transmission bandwidth, each agent compresses its feature map before communication. We employ the variational compression algorithm described in (Ballé et al., 2018) for this purpose, a CNN is trained to compress the feature map, in a way that supports end-to-end optimization. This approach allows the system to preserve essential feature map information while minimizing bandwidth usage. Each agent transmits a compressed form of its intermediate semantic information, denoted as F_N, along with its pose ζ_N, in what we refer to as the collaborative perception message (CPM). This CPM is shared among all neighboring agents. Upon receiving a CPM, decompresses it for further processing, enabling it to select relevant collaborators and transform their feature maps to align with its own perspective.

3.4 Collaborator Selection and Spatial Transformation

It is important to note that not all neighboring agents contribute positively to enhancing the ego agent's situational awareness. In some cases, the semantic information provided by neighboring agents may degrade perception performance due to irrelevant viewpoints (Liu et al., 2020b). Therefore, agent i employs the collaborator selector function which selects only the agents positioned within a 70-meter radius and exhibiting a heading intersection of 70 degrees relative to the ego agent. This relevancy metric range is based on existing dedicated short-range communications (DSRC) standards (Kenney, 2011), and was adopted by multiple collaborative perception methods (Ahmed et al., 2022; Ahmed et al., 2024a; Wang et al., 2020). In this work, we assume ideal communication between agents, where agents consistently transmit and receive the CPMs of their neighbors at each timestep. Since each selected collaborator perceives the environment from different viewpoints and perspectives, its semantic information needs to be transformed to the ego agent's perspective. The ego agent transforms each neighboring agent's semantic information to its perspective using the ego and the selected agents pose ζ_i, ζ_j, respectively. The transformed feature of the j-th agent to the ego agent i is represented as $F_{j \to i} = \Gamma_{j \to i}((F_i, \zeta_i), (F_j, \zeta_i))$, where $\Gamma_{j \to i}$ represents the affine transformation. We utilize the affine transformation due to its ability to preserve parallel lines and distance during rotations. The affine transformation adopted in this work is closely aligned with the method proposed in (Jaderberg et al., 2015), with the key distinction being the absence of a localization network, as each agent broadcasts its pose. The ego agent repeats this affine transformation process for all selected collaborators.

3.5 Graph Fusion Network

Since selected collaborators possess different locations, and viewpoints of the surroundings their semantic information therefore to account for their distinct characteristics, the significance of each agent to the ego agents must be distinguished, and the interactions between multiple agents should vary. To capture this heterogeneity, we present a novel graph iterative attention, employing both spatial and channel attention parallel iteratively to appropriately enhance the feature aggregation. The graph attention-based aggregation scheme proposed indicates i)the collaborator's importance relative to the ego agent, and ii)emphasizing the significant regions within the collaborator's feature map further strengthening the cross-agent feature aggregation.

Graph Network Structure. As shown in Fig. 2, we consider each agent's feature map as a node in the graph, and the edge weights represent the significance of those nodes to each other. Intuitively, we represent the graph as $G = (V, E)$, where V is the set of nodes incorporating the semantic information of each agent $V = \{F_i, F_{j \to i} \dots F_{N \to i}\}$, and E is a set of edges connecting the nodes, where $E = \{W_{ii}, W_{ij}, W_{iN}\}$ represent the importance between selected collaborator and the ego agent determining their significance to each other. In addition to the edge weights W_{iN}, we incorporate an efficient multi-scale attention learning scheme that learns two different attention maps a) spatial attention (α_{sp}) and b) channel attention (α_{ch}) (as shown in Fig. 3). Different from the edge weights that reflect the significance of the nodes to each other, attention directs the models to attend only to significant regions within the feature maps of the selected collaborator relative to the ego agent. Incorporating channel-spatial attention encodes both local and global interactions between connected nodes to better capture the ambiguity in the semantic feature space. Local attention can help preserve object details, while global attention can provide a better understanding of environmental contexts. To this end, we present a graph-structured attention-based fusion process where each agent establishes its own graph, the nodes in the graph maintain the semantic information of the selected collaborators, and the ego node state is updated based on the feature fusion process driver by the edge weights as well the attention maps.

Attention Fusion Module. The attention module includes the parallelly learned channel and spatial at-

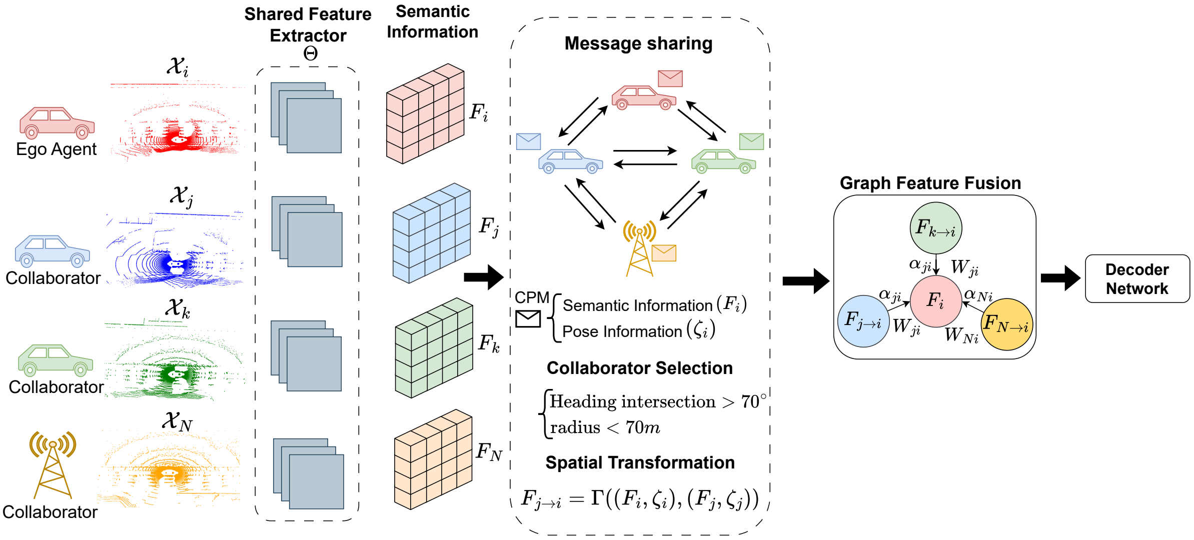

Figure 2: The overall architecture of GIFF. Each agent converts its perceived point cloud $\mathcal{X}_i$ into *BEV* image. The shared feature extractor Θ processes the BEV image to obtain the feature map. Subsequently, each agent shares its CPM to initiate the collaborative selection process. The feature maps corresponding to the relevant agents are then transformed into the ego agent's coordinate system. Subsequently, the features are passed to the graph fusion network to aggregate the collaborator's feature maps with the ego feature map to produce an updated feature representation. The updated representation is then fed to the decoder network to perform object detection.

tention maps as illustrated in Fig. 3. In this manner, the features obtained after applying the attention maps are aggregated to combine both the low and high-level features and effectively direct the attention to the most significant regions within the feature map. Including both channel and spatial attention boosts our proposed fusion strategy to handle objects of varying sizes within the feature maps and aggregate information from multiple receptive fields. Instead of relying solely on global channel attention, which favors large objects, our method incorporates local channel contexts to highlight small objects as well. This allows the network to adaptively adjust its focus based on the scale of the objects present in the image. Additionally, the parallel sub-networks block helps effectively capture the cross-dimension interaction and establish the inter-dimensional dependencies independently. It also allows the information flow within the network by learning which information to emphasize or suppress.

$$F_{ij} = \mathrm{AGG}(F_i, F_{j\to i}) \qquad (1)$$

where AGG is the aggregation operation of the F_i and $F_{j\to i}$, which is computed as summation $F_{j\to i} \in \mathbb{R}^{C\times H\times W}$ or concatenation $F_{j\to i} \in \mathbb{R}^{2C\times H\times W}$ this will be further discussed in section. 6.

- *Channel Attention.* AS shown in Fig. 3(a), to compute the channel attention map α_{ch}, we squeeze the spatial dimension of the aggregated feature F_{ij} by applying global average pooling (GAP) to model only the cross-channel information. GAP generates a compact feature representation F_{ij}^{ch} of shape $\mathbf{R}^{C\times 1\times 1}$ by averaging the spatial dimension within each channel of F_{ij}. This reduces the 2D spatial dimension $H\times W$ into a single value per channel i.e. $C\times 1\times 1$. This distills the most important information from the

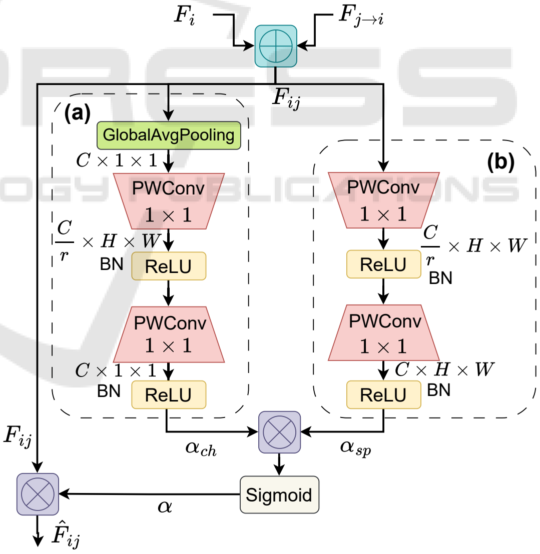

Figure 3: Illustration of the attention map learning scheme within GIFF. Part (a) depicts the channel attention map branch, while part (b) represents the spatial attention map. The parameter r denotes the channel reduction ratio within the encoder-decoder framework.

entire spatial dimension of F_{ij} into a more compact form as illustrated in Fig. 3. F_{ij}^{ch} is then used to learn the per-channel attention map that reflects how important each channel is for the object detection task. To learn the attention map, and maintain a lightweight model, F_{ij}^{ch} is passed to an encoder-decoder point-wise convolution network (PwConv) (1×1) which local channel context aggregator exploiting only channel interactions. The channel attention map α_{ch} is learned as follows:

$$F_{ij}^{ch} = \mathrm{GAP}(F_{ij}) \tag{2a}$$

$$\alpha_{ch} = \Psi(\Upsilon(F_{ij}^{ch})) \tag{2b}$$

Ψ and Υ are the decoding-encoding PwConv-based network used to learn the channel attention map.

- *Spatial Attention.* In parallel, we generate a spatial attention map by utilizing the spatial relationship of features as shown in Fig. 3(b). Different from channel attention, spatial attention focuses on where within the spatial dimension are the informative parts of the aggregated feature F_{ij}, and increases their weight within the attention map. On F_{ij}, we apply encoder-decoder PwConv layers to generate the spatial attention map α_{sp}. The spatial attention process is expressed as follows:

$$\alpha_{sp} = \Omega(\Lambda(F_{ij})) \tag{3}$$

where Ω and Λ are the decoding-encoding PwConv-based network tailored to learn the spatial attention map.

Following the computation of the channel and spatial attention maps, these maps are combined to form the final feature map, denoted as α, in order to exploit the learned representations. The feature map α is subsequently utilized to update the aggregated feature map F_{ij}, resulting in the refined feature map $F_{ij}^{(l)}$, as expressed by:

$$\alpha = \sigma(\alpha_{ch} \oplus \alpha_{sp}) \tag{4}$$

$$\hat{F}_{ij} = \alpha \otimes F_{ij} \tag{5}$$

- *Iterative Attention* To enhance the attention map of complementary information from the aggregated features, we propose an iterative attention learning strategy. This strategy progressively refines the spatial and channel attention maps, thereby enhancing the discriminative power of the learned features and improving the feature fusion process. At the end of each iteration, the features are aggregated and forwarded to the next iterative layer, refining the input to the attention module and potentially generating more expressive attention maps. After l iterations, as illustrated in Fig. 4, the fused feature $F_{ij}^{(l)}$ is incrementally updated through each attention iteration, ultimately producing the final representation $\hat{F}_{ij}$.

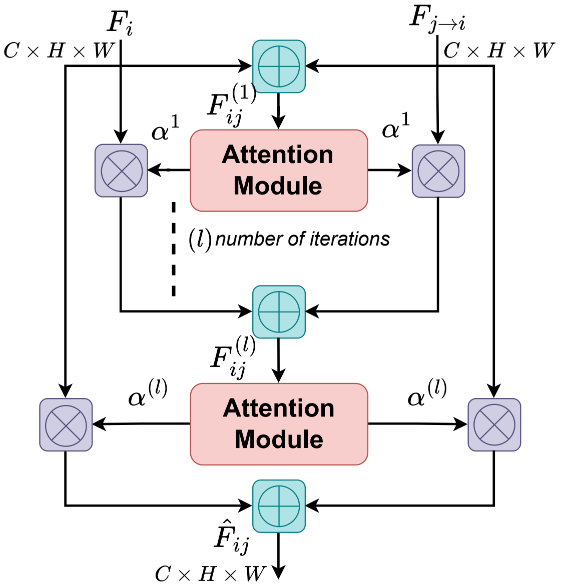

Figure 4: Illustration of the proposed iterative attention feature fusion where the attention module, shown in Fig.3, is repeated for l iterations.

The attention fusion module is repeated for every connected node to compute the updated feature $\hat{F}_{iN}$.

Node Feature Aggregation. After obtaining updated features from the attention fusion module, the final updated feature, H_i is computed as a weighted sum, where each feature $\hat{F}_{iN}$ s multiplied by its corresponding learnable edge weight matrix W_{iN} as follows:

$$H_i = \sum(W_{iN}\hat{F}_{iN}) \tag{6}$$

3.6 Decoder Network

After the graph-based fusion, the ego agent the feature map H_i is passed into the detection decoder that decodes it into objects, including class and regression output. This study aims to enhance the feature aggregation methodology using a graph attention-based network and assess its performance in comparison to state-of-the-art techniques. In line with the feature extractor (discussed in Section 3.2, we adopt the same detection decoder network utilized by DiscoNet $\Phi_{\text{DiscoNet}}(\cdot)$ (Li et al., 2021), and V2VNet $\Phi_{\text{V2VNet}}(\cdot)$ (Wang et al., 2020), to produce the final detection outputs.

4 EXPERIMENTAL SETUP

Dataset. We evaluate our work using V2X-Sim (Li et al., 2022) and OPV2V (Xu et al., 2022b) datasets. **V2X-Sim** dataset integrates the SUMO platform (Krajzewicz et al., 2012) for generating traffic flow data and the Carla simulator (Dosovitskiy et al., 2017) to capture sensor data from multiple agents. V2X-Sim consists of 10,000 frames across 100 scenes, each involving 2-5 collaborative agents. We split the dataset into training, validation, and test sets containing 8,000, 1,000, and 1,000 frames, respectively. Each frame includes data collected from vehicles and roadside units (RSUs), resulting in 37,200 training samples, 5,000 validation samples, and 5,000 test samples. This work evaluates object detection performance in two scenarios: without RSU (w/o RSU) and with RSU (w/ RSU).

OPV2V is a large-scale V2V perception dataset created utilizing CARLA (Dosovitskiy et al., 2017) and OpenCDA (Xu et al., 2021). The dataset consists of around 11,464 LiDAR point cloud frames. OPV2V is divided into two subsets: the default CARLA towns and the Culver City digital town. The default town subset has a total of 10,914 frames. These frames are divided into train/val/test splits of 6,764/1,980/2,170 frames, respectively. This subset offers a broad spectrum of scenarios characterized by varying levels of complexity. In contrast, the Culver City subset consists of 550 frames used for evaluation that simulate a real-world urban environment, with a wide range of objects and structures.

Evaluation Metrics. To supervise foreground-background classification loss, we utilize the binary cross-entropy (Mannor et al., 2005). For the bounding-box regression loss, we utilize the weighted smooth loss. To assess the collaborative perception detection performance we utilize average precision (AP) over the Intersection over Union (IoU) thresholds of 0.5 and 0.7.

Training Setup. We utilize the Adam optimizer with an initial learning rate of 10^{-3} and steadily decay at every 10 epochs using a factor of 0.1. All models are trained on NVIDIA Tesla V100 GPU with a batch size of 4. We compare GIFF with no, early, and late collaboration methods. For the intermediate collaboration methods, we benchmark six approaches that evaluated their result using V2XSim: When2Com (Liu et al., 2020a), Who2Com (Liu et al., 2020b), V2VNet (Wang et al., 2020), DiscoNet (Li et al., 2021), Ahmed et. al. (Ahmed et al., 2024b), CollabGAT (Ahmed et al., 2024a). For OPV2V the benchmarks are: F-Cooper(Chen et al., 2019), Who2Com, AttFuse (Xu et al., 2022b), V2VNet, HP3D-V2V (Chen et al., 2024) and CollabGAT.

Table 1: Object detection AP on V2X-SIM reporting results of both with and without RSU at IoU of 0.5 and 0.7. Note results in **red**, **blue**, **green** denoting the 1^{st},2^{nd} and 3^{rd} highest AP results.

Method	AP@IoU=0.5		AP@IoU=0.7	
	w/o RSU	w/RSU	w/o RSU	w/RSU
When2com	44.02	46.39	39.89	40.32
Who2com	44.02	46.39	39.89	40.32
V2VNet	68.35	72.08	63.83	65.85
DiscoNet	**69.03**	72.87	63.44	**66.40**
Ahmed et. al	68.97	**72.96**	**63.48**	65.94
CollabGAT	**69.67**	**75.57**	**63.72**	**73.29**
GIFF (Ours)	**73.62**	**78.93**	**68.37**	**75.82**
No Collaboration	49.90	46.96	44.21	42.33
Late Collaboration	43.99	42.98	39.10	38.26
Early Collaboration	70.43	77.08	67.04	72.57

Table 2: Object detection AP on OPV2V reporting results tested on default and Culver at IoU of 0.5 and 0.7. Note results in **red**, **blue**, **green** denoting the 1^{st},2^{nd} and 3^{rd} highest AP results.

Method	Default		Culver	
	AP@0.5	AP@0.7	AP@0.5	AP@0.7
F-Cooper	61.77	49.85	53.79	44.50
Who2Com	62.04	50.52	54.11	44.21
AttFuse	62.86	50.84	54.01	46.37
V2VNet	63.33	51.67	54.54	45.87
HP3D-V2V	**67.42**	**56.50**	**58.83**	**50.51**
CollabGAT	**68.41**	**58.32**	**60.01**	**51.82**
GIFF (Ours)	**69.60**	**60.04**	**61.35**	**51.93**
No Collaboration	49.13	38.38	40.66	26.70
Late Collaboration	59.61	42.53	49.45	39.76
Early Collaboration	52.35	40.66	42.59	35.34

5 RESULTS AND DISCUSSION

Detection Performance. Tables. 1 and 2 shows the AP object detection performance of GIFF on V2XSim and OPV2V datasets. As shown in Table. 1, our method significantly outperforms V2VNet (Wang et al., 2020), DiscoNet (Li et al., 2021), (Ahmed et al., 2024b), and CollabGAT (Ahmed et al., 2024a); for instance, at IoU of 0.7 w/RSU, our method achieves performance gains of 15.14%, 14.19%, 15%, and 3.45%, respectively. For the OPV2V results shown in Table. 2, among all fusion models GIFF consistently achieves the highest AP scores for both driving scenarios. Especially GIFFS's superiority in the Culver City scenario demonstrates its strong generalization ability. These illustrated results highlight the effectiveness of GIFF in enhancing the object detection AP when compared to other state-of-the-art intermediate collaboration methods. This improvement can be attributed to our proposed iterative attention-based learning network, which iteratively refines the attention map, allowing the model to focus more accurately on relevant regions in both ego-centric and received semantic information. Unlike the attention mechanisms in CollabGAT and Ahmed et al., which

Table 3: Number of parameters of each model trained on V2XSim dataset.

Method	No. of parameters (M)
V2VNet	21.08
DiscoNet	15.84
Ahmed et. al	15.98
CollabGAT	15.93
GIFF (Ours)	16.12

also incorporate attention within their method, our iterative attention fusion approach enables superior feature fusion by progressively learning the significance of each feature map in relation to ego-centric semantic information while preserving spatial relationships across feature maps. In addition to that, the attention cooperation within the multi-agent fusion scheme, where channel attention directs the model to relevant features across channels, while spatial attention focuses on important spatial locations, enhancing the model's overall feature fusion.

Computational Efficiency. Table. 3 presents the parameter counts for each state-of-the-art method. Our proposed method demonstrates a 23.5% reduction in parameter count compared to the V2VNet network. For other methods, the parameter count of our model is either comparable or marginally higher, with an increased range of approximately 0.9% to 1.7%. However, this slight increase is negligible given the substantial performance improvements achieved. This efficiency is attributed to our iterative PWConv attention mechanism, which iteratively enhances the attention map without significantly impacting model size, thereby supporting performance gains in object detection. A more detailed analysis is presented in the ablation study (Section 6).

Table 4: This table gives an experiment number to differentiate the different settings of GIFF conducted within the ablation study.

Experiment No.	Model Base	Aggregation Operation	Depth
1	**DiscoNet**	Sum	256, 128, 64
2(Default)		Sum	256, 128, 64, 32
3		Concat	512, 256, 128, 64
4		Concat	512, 256, 128, 64, 32
5	**V2VNet**	Concat	512, 256, 128, 64
6		Concat	512, 256, 128, 64, 32

6 ABLATION STUDY

Table. 4 shows the design of each experiment conducted to evaluate the effect of every module of GIFF, with every design carrying the species experiment tag.

Effect of Deeper Attention Layers. This module is defined by Eqs. 2b and 3, which govern the learning of channel and spatial attention maps. As shown in Table 5, a deeper encoder-decoder architecture leads to a higher object detection AP. This improvement is attributed to our proposed attention network, which is based on a PwC framework. In this network, deeper layers capture higher-level and more abstract representations of the input data. Consequently, the network learns intricate patterns and correlations among features, as the deeper layers combine features learned in earlier stages to create representations that capture more complex aspects of the input. These high-level representations are crucial for learning attention weights effectively. However, we observed that increasing depth beyond the tested level led to a decrease in AP due to the vanishing gradient problem, where the gradient signal becomes too weak to propagate effectively through multiple layers.

Effect of Aggregation Operation. This section examines the aggregation function "AGG" employed in Eq. 1. As presented in Table 5, the experimental setup in experiment "2" achieves the highest detection AP with a minimal model parameter count, while experiment "1" attains the second-highest AP, however, it achieves the lowest parameter count of all experiments. Quantitative analysis of the proposed methodology demonstrates that summation slightly outperforms concatenation. This can be attributed to summation's ability to seamlessly integrate information, effectively combining low-level details (such as edges) with high-level semantics (such as object shapes), thus yielding more cohesive and generalizable features. Additionally, summation aids gradient flow during backpropagation by preserving feature map size and channel consistency, which contributes to stable training—especially in deep networks prone to gradient degradation. Concatenation increases dimensionality and computational requirements, as reflected in a higher parameter count in the last column of Table 5.

Effect of Iterative Fusion. As shown in Table. 5 adding another layer of attention further improves the performance as the iterative extraction allows the model to tune the parameters to extract even more information from the initially fused feature map. However, this improvement may be obtained at the cost of increasing the model's number of parameters. Interestingly, we find that extra iterations do not boost performance, and two iterations achieve the best results in our experiment.

Table 5: The AP and the number of parameters are represented by different design considerations of GIFF. Aggregation Operation represents the "AGG" featured in Eq. \ref{aggregate_features}. "Depths" represents the feature map dimensionality reduction to compute the attention weights. w/IAtten and w/o represent the AP with and without iterative attention.

Experiment No.	AP@IoU=0.5				AP@IoU=0.7				No. of Parameters (M)	
	w/ IAtten		w/o IAtten		w/ IAtten		w/o IAtten		w/ IAtten	w/o IAtten
	w/o RSU	w/RSU	w/o RSU	w/RSU	w/o RSU	w/RSU	w/o RSU	w/RSU		
1	71.28	77.26	68.56	71.81	66.85	73.29	62.88	65.86	16.10	15.39
2 (Default)	**73.62**	**78.93**	68.97	72.96	**68.37**	**75.82**	63.48	65.94	**16.12**	15.98
3	68.53	74.33	68.14	71.61	62.76	67.42	62.36	64.41	16.89	16.20
4	67.15	72.14	68.50	72.25	64.32	68.46	63.32	63.74	16.92	16.21
5	68.56	70.05	67.53	70.0	62.68	71.14	61.55	63.52	17.06	16.78
6	69.93	72.78	68.46	70.94	63.12	68.37	63.10	63.10	17.13	16.85

7 CONCLUSION AND FUTURE WORK

This paper presents GIFF, a graph iterative attention-based network designed to address collaborative perception challenges in multi-agent systems. GIFF effectively facilitates multi-agent collaboration by intelligently fusing perceptual information received from collaborators. It achieves this by learning the relative importance of collaborators and identifying the spatial regions within the received semantic information that require higher attention. The iterative attention mechanism further enhances the refinement of the attention-learning process. GIFF achieves superior performance on the object detection task, as demonstrated on standard benchmarks such as V2XSim and OPV2V. Despite these promising results, the approach has significant potential for future improvements. As part of future work, we aim to address the impact of transmission delays caused by communication network characteristics, which hinder the performance of collaborative perception.

REFERENCES

Ahmed, A. N., Anwar, A., Mercelis, S., Latré, S., and Hellinckx, P. (2021). Ff-gat: Feature fusion using graph attention networks. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6. IEEE.

Ahmed, A. N., Mercelis, S., and Anwar, A. (2024a). Collabgat: Collaborative perception using graph attention network. *IEEE Access*.

Ahmed, A. N., Mercelis, S., and Anwar, A. (2024b). Graph attention based feature fusion for collaborative perception. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2317–2324. IEEE.

Ahmed, A. N., Ravijts, I., de Hoog, J., Anwar, A., Mercelis, S., and Hellinckx, P. (2022). A joint perception scheme for connected vehicles. In *2022 IEEE Sensors*, pages 1–4. IEEE.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*.

Chen, H., Wang, H., Liu, Z., Gu, D., and Ye, W. (2024). Hp3d-v2v: High-precision 3d object detection vehicle-to-vehicle cooperative perception algorithm. *Sensors*, 24(7):2170.

Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., and Fu, S. (2019). F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R. R., Cheng, M.-M., and Hu, S.-M. (2022). Attention mechanisms in computer vision: A survey. *Computational visual media*, 8(3):331–368.

Han, Y., Zhang, H., Li, H., Jin, Y., Lang, C., and Li, Y. (2023). Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*.

Hu, Y., Fang, S., Lei, Z., Zhong, Y., and Chen, S. (2022). Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in neural information processing systems*, 28.

Kenney, J. B. (2011). Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4).

Li, Y., Ma, D., An, Z., Wang, Z., Zhong, Y., Chen, S., and Feng, C. (2022). V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921.

Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., and Zhang, W. (2021). Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552.

Liu, Y.-C., Tian, J., Glaser, N., and Kira, Z. (2020a). When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4106–4115.

Liu, Y.-C., Tian, J., Ma, C.-Y., Glaser, N., Kuo, C.-W., and Kira, Z. (2020b). Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6876–6883. IEEE.

Mannor, S., Peleg, D., and Rubinstein, R. (2005). The cross entropy method for classification. In *Proceedings of the 22nd international conference on Machine learning*, pages 561–568.

Qian, R., Lai, X., and Li, X. (2022). 3d object detection for autonomous driving: A survey. *Pattern Recognition*, 130:108796.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., and Urtasun, R. (2020). V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 605–621. Springer.

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., and Ma, J. (2021). Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE.

Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.-H., and Ma, J. (2022a). V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pages 107–124. Springer.

Xu, R., Xiang, H., Xia, X., Han, X., Li, J., and Ma, J. (2022b). Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Zhou, Y., Xiao, J., Zhou, Y., and Loianno, G. (2022). Multi-robot collaborative perception with graph neural networks. *IEEE Robotics and Automation Letters*, 7(2):2289–2296.