# Collaborative Decision-Making Using Spatiotemporal Graphs in Connected Autonomy

Peng Gao, Yu Shen, and Ming C. Lin

***Abstract*— Collaborative decision-making is an essential capability for multi-robot systems, such as connected vehicles, to collaboratively control autonomous vehicles in accident-prone scenarios. Under limited communication bandwidth, capturing comprehensive situational awareness by integrating connected agents' observation is very challenging. In this paper, we propose a novel collaborative decision-making method that efficiently and effectively integrates collaborators' representations to control the ego vehicle in accident-prone scenarios. Our approach formulates collaborative decision-making as a classification problem. We first represent sequences of raw observations as spatiotemporal graphs, which significantly reduce the package size to share among connected vehicles. Then we design a novel spatiotemporal graph neural network based on heterogeneous graph learning, which analyzes spatial and temporal connections of objects in a unified way for collaborative decision-making. We evaluate our approach using a high-fidelity simulator that considers realistic traffic, communication bandwidth, and vehicle sensing among connected autonomous vehicles. The experimental results show that our representation achieves over 100x reduction in the shared data size that meets the requirements of communication bandwidth for connected autonomous driving. In addition, our approach achieves over 30% improvements in driving safety.**

## I. Introduction

Multi-robot systems have received considerable attention over the past few decades, due to their remarkable attributes of redundancy [1], scalability [2], and parallelism [3]. Among these, the connected autonomous vehicle stands out as a prominent example of collaborative multi-robot systems. Unlike conventional studies that focus on single-robot scenarios, connected autonomous driving considers the collective capabilities of multiple autonomous vehicles, enabling enhanced performance and efficiency in diverse tasks, such as object detection [4], tracking [5] and autonomous driving control [6]

To enable efficient connected autonomous driving, collaborative decision-making is a fundamental ability, with the goal of enabling connected vehicles to efficiently share and utilize observations provided by ego and collaborator vehicles, thus mitigating blind spots and collaboratively making optimal decisions (e.g., taking brake actions), especially in accident-prone scenarios. As shown in Figure, the ego vehicle (shown in gray) is blocked by yellow cars, and it may be difficult for the car's sensors to detect an approaching red car rushing through the intersection. In this case, by integrating observations provided by collaborative vehicles, the ego vehicle eliminates its own blind spots and takes a "brake" action to avoid a traffic accident. However, collaborative decision-making in connected autonomous driving is very challenging, as the communication bandwidth between vehicles is limited, which does not allow connected vehicles to directly share their raw observations [7], [8]. In addition, occlusion in observations, highly dynamic street conditions, and the complex interactions of objects make collaborative decision-making very hard to solve.

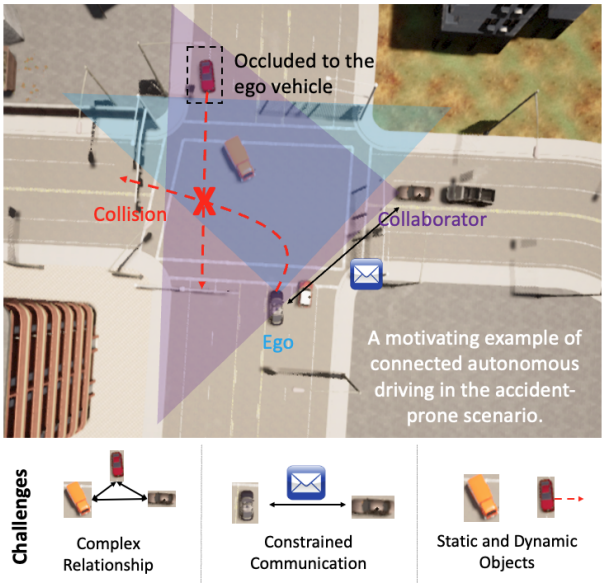


Fig. 1. An motivating example of connected autonomous driving in the accident-prone scenario. To avoid accidents, connected vehicles need to effectively share and integrate their own observations, meanwhile addressing the challenges caused by communication constraint and complex interactions among static and dynamic street objects.

Various methods are studied to address the challenges, which can be generally divided into three groups, including raw-based early collaboration, output-based late collaboration, and feature-based intermediate collaboration methods. Raw-based early collaboration utilizes multi-robot raw observations, which generates the most comprehensive situational awareness. These methods can effectively remove blind spots of individuals, but far beyond communication bandwidth limitation in outdoor applications [9]. Output-based late collaboration methods focus on sharing and fusing multi-robot prediction results [10], [11], which significantly reduces the consumption of communication bandwidth. However, these results are predicted based on single-robot observations with incomplete information about a scene, which are highly to be inaccurate and noisy. Feature-based intermediate collaboration methods mainly focus on utilizing the extracted fea-

Peng Gao, Yu Shen, and Lin C. Lin are with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Lin is also with Amazon. Email: {gaopeng, yushen, lin}@umd.edu.

tures for collaboration tasks, such as sharing downsampling lidar points [4] or extracting features via PointNet [6]. These approaches achieve a trade-off between communication efficiency and performance. However, how to compactly preserve the cues of a scene for sharing and aggregation, such as temporal cues, has not been well addressed yet.

In this paper, we propose a novel collaborative decision-making method that efficiently integrates spatiotemporal observations provided by connected vehicles for autonomous driving. First, we represent each vehicle's observation sequence as a *spatiotemporal graph*, with the nodes to encode the locations of the detected objects, the spatial edges to encode the spatial distance between pairs of objects, and temporal edges to encode the motion of objects. Given the spatiotemporal representations, connected vehicles can efficiently share sequential observations while overcoming communication constraints. Then, we merge all ego-collaborator representations given the Global Navigation Satellite System (GNSS) poses and formulate collaborative decision-making as a classification problem. We utilize *heterogeneous graph learning* as a framework that simultaneously analyzes spatial and temporal relationships of objects, thus encoding comprehensive situational awareness. Given the situational embedding, we predict if the ego vehicle should take brake action to avoid an accident or not. The full approach is learned by imitation learning with expert actions.

Our key contribution is the introduction of a novel collaborative decision-making method for connected autonomous driving. Specifically,

- We propose a novel representation based on *spatiotemporal graphs* generated from a sequence of observations, which integrate not only the current states but also the historical motion of street objects. Our representation achieves a greater than **100x** reduction in the shared data size that meets the requirements of communication bandwidth for connected autonomous driving.
- We present a novel *spatiotemporal graph neural network* based on *heterogeneous graph learning*, which generates the embedding of the spatiotemporal graph by analyzing the spatial and temporal connection of objects in a unified way, thus encoding cues for collaborative decision-making. Our approach achieves over **30**% improvements in driving safety.

## II. Related Work

Connected autonomous driving based on collaborative perception provided by connected agents has attracted extensive attention recently. The existing methods can be generally divided into three groups, including raw-based early collaboration, output-based late collaboration, and feature-based intermediate collaboration methods.

The early collaboration fuses the raw data for the input of the network, which requires connected agents to share, transform, and aggregate raw sensor data onboard for vision task [12], [13]. The late collaboration usually adopts fusion at the postprocessing stage, which merges multi-agent perception outputs, such as Non-Maximum suppression to remove redundant prediction [10] and refined matching to remove results that violate the pose consistency [11]. The intermediate collaboration aims to learn and share compressed features from the raw observations, which is a trade-off between communication bandwidth and performance. From the data-sharing perspective, different communication mechanisms are developed, such as When2com [14], Who2com [15], and Where2com [16]. From the data fusion perspective, the strategies include direct concatenation [17], re-weighted sum [18], graph learning-based fusion [19], [20], and attention-based fusion [21], [22]. From the task perspective, various tasks are studied, including object detection [23], tracking [5], semantic segmentation [24], localization [25], depth estimation [26] and autonomous driving control [6].

The early collaboration contains the most comprehensive information of a scene, which can overcome occlusion and long-distance observations, but these methods can not be used in outdoor environments due to their large bandwidth requirements. Late collaboration significantly reduces the communication bandwidth cost by directly sharing outputs, however, as individual observations are often noisy or incomplete, late collaboration generally has the worst performance. Intermediate collaboration balances communication efficiency and collaboration performance. However, these methods do not have redundant bandwidth to share a sequence of features, which causes the ignoring of temporal cues for collaborative decision-making.

To encode spatial and temporal information for decision-making, spatiotemporal graph learning is widely used in single-robot scenarios, such as trajectory prediction [27], path planning [28], object localization [29], [30], and reasoning [31]. These methods generally treat the embedding of the spatial and temporal domain separately, such as using a recurrent neural network to aggregate temporal information and using a graph neural network to aggregate spatial information [32], [32], [33], or alternating temporal and spatial blocks during convolution [34], [35], [36]. Even though these approaches have achieved promising results in vision tasks, they typically assume that a person or vehicle can be continuously tracked, and use a fixed-length observation sequence as input. Therefore these methods are difficult to apply to collaborative decision-making with the observed objects being prone to intermittent losses within the observation sequence and the available sequence length being variant. In this paper, we propose a novel spatiotemporal graph network based on heterogeneous graph learning, which can deal with length-variant observations with objects missing, meanwhile preserving spatial and temporal connections of objects for connected autonomous driving.

## III. Approach

**Notation.** Matrices are represented as boldface capital letters, e.g., $\mathbf{M} = \{\mathbf{M}_{i,j}\} \in \mathcal{R}^{n \times m}$. $\mathbf{M}_{i,j}$ denotes the element in the $i$-th row and $j$-th column of $\mathbf{M}$. Vectors are denoted as boldface lowercase letters $\mathbf{v} \in \mathcal{R}^n$ and scalars are denoted as lowercase letters.

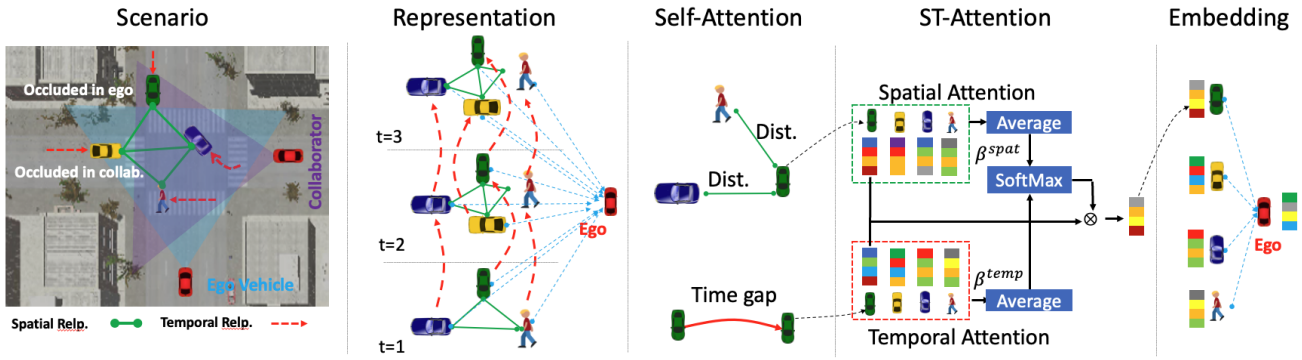


Fig. 2. An overview of our approach of spatiotemporal graph embedding for connected autonomous driving. An ego vehicle and a collaborator vehicle meet at an intersection, each of them has occluded objects in their own field of view. We represent their observation sequences as spatiotemporal graphs and merge them all to generate the final representation of the scene. Our spatiotemporal graph learning network captures spatial and temporal relationships of objects by analyzing the importance of spatial and temporal attention and, finally, generating the situational embedding of the scene with respect to the ego vehicle.

### A. Problem Formulation

We propose a collaborative decision-making approach for connected autonomous driving in accident-prone scenarios. Formally, we assume that there is one ego vehicle and $n-1$ collaborative vehicles that receive RGB-D or Lidar observations. Each vehicle provides a sequence of observations $\mathcal{O}_k = \{obs_k^t, obs_k^{t+1}, \ldots, obs_k^{t+T}\}, k \in \{0, 1, 2, \ldots, n\}$. Each observation recorded at time $t$ consists of detected objects $obs^t = \{\mathbf{v}_1^t, \mathbf{v}_2^t, \ldots, \mathbf{v}_m^t\}$ where $\mathbf{v}_i \in \mathbb{R}^3$ denotes the 3D position of the $i$-th object detected at time $t$.

Given a sequence of observations $\mathcal{O}_k$, we represent it as a spatiotemporal graph $\mathcal{G}_k = \{\mathcal{V}_k, \mathcal{E}_k^{spat}, \mathcal{E}_k^{temp}\}, k = 1, 2, \ldots, n$. $\mathcal{V}_k = Unique(\mathcal{O}_k)$ denotes the node set, which contains all the positions of objects detected by $n$ vehicles, where $Unique$ denotes the unique operation that removing the duplicated objects in $\mathcal{O}_k$. In addition, $\mathcal{E}_k^{spat} = \{e_{p,q}^{spat}\}$ denotes the spatial relationships between a pair of objects. $e_{p,q}^{spat} = ||\mathbf{v}_p^t - \mathbf{v}_q^t||_2$ denotes the distance between the $p$-th object and the $q$-th object recorded at the same time $t$, otherwise $e_{p,q}^{spat} = 0$. $\mathcal{E}_k^{temp} = \{e_{p,q}^{temp}\}$ denotes the temporal relationships of the same object recorded at different times. If $\mathbf{v}_p^{t_1}$ and $\mathbf{v}_q^{t_2}$ are the same object recorded at time $t_1$ and $t_2$, then $e_{p,q}^{temp} = t_2 - t_1$, otherwise $e_{p,q}^{temp} = 0$.

Given the $n$ spatiotemporal graph representations provided by $n$ connected vehicles, we first transform the collaborator vehicles' observations to the ego vehicle's coordinate and merge them together with the ego observations, thus eliminating the ego vehicle's blind spots. The transformation of each collaborative vehicle is obtained from the GNSS sensor [6]. The merged spatiotemporal graph is denoted as $\mathcal{M} = \phi(\mathcal{G}_{ego}, \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m)$, where $\phi$ denotes the transformation and merge function and $m$ denotes the number of collaborator vehicles that are close to the ego vehicle within a distance threshold. The threshold is set to 150 meters. As the ego vehicle may be observed by other vehicles, we add the ego node $obs_{ego} = [0, 0, 0]$ and remove all nodes within 2 meters away from the ego node. Then, we fully connect the ego node to all the other nodes for message-passing purposes. Finally, the merged spatiotemporal graph is defined as $\mathcal{M} = \{\mathcal{V}, \mathcal{E}^{spat}, \mathcal{E}^{temp}\}$, as shown in Figure 2.

Based upon the merged spatiotemporal graph $\mathcal{M}$, we formulate collaborative decision-making for connected autonomous driving as a classification problem. The goal is to identify if the current situation is dangerous or not, thus allowing the ego vehicle to take brake or driving actions. Given the spatiotemporal graph representations, we significantly reduce the amount of shared data packages between connected vehicles and preserve the important cues, including spatial and temporal relationships of street objects, for decision-making in accident-prone scenarios.

### B. Spatiotemporal Graph Embedding

To obtain the embedding of spatiotemporal graphs for decision-making, we propose a heterogeneous graph attention network that encodes object locations and their spatiotemporal relationships in a unified way, as shown in Figure 2. The embedding of the spatiotemporal graph is defined as $\mathbf{h}'_{ego} = \psi(\mathcal{M})$, where $\mathbf{h}'_{ego}$ is the embedding of the holistic scene with respect to the ego vehicle and $\psi$ is the heterogeneous attention network. Formally, we first project each node feature to the same feature space, which is defined as follows:

$$\mathbf{h}_i = \mathbf{W}_v \mathbf{v}_i \tag{1}$$

where $\mathbf{h}_i$ denote the projected feature of the $i$-th node, $\mathbf{W}_v$ denote the associating weight matrix. $\mathbf{v}_i$ denotes the position of the $i$-th object. Then we compute the self-attention of each node given the spatial edges, defined as follows:

$$\alpha_{i,j}^{spat} = \frac{\exp\left(\sigma([\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j || \mathbf{W}_e e_{i,j}])\right)}{\sum_{e_{i,k} \in \mathcal{E}^{spat}} \exp(\sigma([\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_k || \mathbf{W}_e e_{i,k}]))} \tag{2}$$

where $\alpha_{i,j}^{spat}$ is the attention from node $j$ to node $i$, $\sigma$ denotes the ReLu activation function, $||$ denotes the concatenation operation, $\mathbf{W}$ and $\mathbf{W}_e$ are weight matrices. This attention weight is obtained by comparing the query of the $i$-th node with its neighborhood nodes meanwhile considering their edge attributes. The final attention is normalized by the SoftMax function. To encode temporal relationships of objects, we can easily traverse all temporal edges in $\mathcal{E}^{temp}$ in Eq. (2). Finally, given different types of edges, we get two attentions $\alpha_{i,j}^{spat}, \alpha_{i,j}^{temp}$ for the $i$-th node. Then, we compute the node embedding vector as follows:

$$\mathbf{h}_i^{spat} = \sigma\left(\mathbf{W}\mathbf{h}_i + \sum_{e_{i,k}\in\mathcal{E}^{spat}} \alpha_{i,j}^{spat}(\mathbf{W}\mathbf{h}_j + \mathbf{W}_e e_{i,j})\right) \tag{3}$$

Based upon Eq. (3), we compute the $i$-th node embedding vectors $\mathbf{h}_i^{spat}$ and $\mathbf{h}_i^{temp}$ given their associating attentions $\alpha_{i,j}^{spat}, \alpha_{i,j}^{temp}$. They are computed via aggregating the object embedding feature and their spatiotemporal edge attributes weighted by attention weights. We also use a multi-head mechanism to enable the network to catch a richer representation of the embedding. Multi-head embedding vectors are concatenated after intermediate attention layers.

Given the spatiotemporal embedding of nodes, we further learn the importance of different types of relationships of objects. The importance of spatial relationships is computed as follows:

$$\beta^{spat} = \frac{1}{|\mathcal{V}^{spat}|}\sum_{i\in\mathcal{V}^{spat}} \mathbf{q}^T tanh(\mathbf{W}_b\mathbf{h}_i^{spat} + \mathbf{b}) \tag{4}$$

where $\mathbf{W}_b$ denotes the weight matrix, $\mathbf{b}$ is the bias vector, $\mathbf{q}$ denotes the learnable edge-specific attention vector and $\mathcal{V}^{spat}$ denotes the node set containing nodes with spatial edge connections. The importance of spatial relationships is obtained by averaging the spatial embedding vectors of all nodes. Similarly, the importance of temporal relationships is computed through Eq. (4) with the node set $\mathcal{V}^{temp}$, where $\mathcal{V}^{temp}$ denotes the node set containing nodes with temporal edge connections. The learnable parameters are shared for the computation of the importance of spatial and temporal relationships. Then, $\beta^{spat}$ and $\beta^{temp}$ are normalized through SoftMax, defined as follows:

$$\beta^{spat} = \frac{\exp(\beta^{spat})}{\exp(\beta^{spat} + \beta^{temp})} \tag{5}$$

$$\beta^{temp} = \frac{\exp(\beta^{temp})}{\exp(\beta^{spat} + \beta^{temp})} \tag{6}$$

where $\beta^{spat}, \beta^{temp}$ denotes the contribution of the type of relationship for decision-making. The higher the value, the larger the importance of the type of relationship. The final situational embedding vector with respect to the ego vehicle is computed as:

$$\mathbf{h}'_{ego} = \sum_{\Psi\in\{spat,temp\}} \beta^{\Psi}\mathbf{h}_{ego}^{\Psi} \tag{7}$$

where $\mathbf{h}'_{ego}$ denotes the node embedding vector of the ego node, which integrates all object positions and their spatiotemporal relationships.

### C. Connected Autonomous Driving

Given the status embedding of the ego vehicle $\mathbf{h}'_{ego}$, we predict the action of the ego vehicle to identify if the ego vehicle should take brake action or not. Formally, the prediction is defined as follows:

$$\mathbf{p} = \text{SoftMax}\left(\sigma\left(MLP([\mathbf{h}'_{ego}||cmd])\right)\right) \tag{8}$$


Fig. 3. Three scenarios of CAD, including overtaking, left turn, and street crossing. Each scenario contains one ego vehicle and at least two collaborator vehicles.

TABLE I

COMMUNICATION STANDARD USED IN THE SIMULATION WITH THE SENSING FREQUENCY OF 10 HZ

| Method | Bandwidth | Package Size | Packet Loss |
|---|---|---|---|
| DSRC [7] | 2 Mbps | 200KB | $< 5\%$ |
| C-V2X [8] | 7.2 Mbps | 720 KB | $< 5\%$ |

where $\mathbf{p} = [p_1, p2]$ denotes the prediction with $p_1$ denoting the probability of taking brake and $p_2$ denoting the probability of keeping running, and $p_1 + p_2 = 1$. $cmd$ denotes the command given to the ego vehicle, including lane follows, turn right, turn left, go straight, change left, and change right, which is represented as a one-hot vector. $\sigma$ denotes the $ReLu$ activation function and $MLP$ denotes a multi-layer perceptron which contains one linear layer. Finally, the prediction result is normalized by the SoftMax function. To train our network, we use the cross-entropy loss.

## IV. EXPERIMENT

### A. Experimental Setups

We employ a high-fidelity connected autonomous driving (CAD) simulator to evaluate our approach, which integrates CARLA [39] and AutoCast [40]. CARLA is an open-source autonomous driving simulator that is able to simulate vehicle sensors and traffic scenarios. AutoCast is an end-to-end framework built upon CARLA, which provides V2V communication and sensor sharing, thus achieving collaborative decision-making and vehicle collaboration. In our experiments, following the recent work [6], we utilize three different traffic scenarios at street interactions where accidents more frequently occur, as depicted in Figure 3. Specifically,

- **Overtaking**: A truck is obstructing a sedan's path on a two-way, single-lane road marked with a dashed yellow divider. Moreover, the truck is hindering the sedan's visibility of the oncoming lane. The autonomous vehicle (ego car) needs to perform a lane change maneuver to pass the truck.
- **Left Turn**: An ego car attempts to make a left turn at a yield light. However, it faces an obstacle in the form of another truck positioned in the opposing left-turn lane. This obstacle limits the ego car's visibility of the

TABLE II
QUANTITATIVE COMPARISON OF DIFFERENT METHODS FOR CONNECTED AUTONOMOUS DRIVING: ACCIDENT DETECTION (**AD**) FOR DRIVING SAFETY AND EXPERT ACTION RATE (**EAR**) FOR THE IMITATION PERFORMANCE.

| Method | Package Size | Overtaking | | Left Turn | | Street Crossing | |
|---|---|---|---|---|---|---|---|
| | PS ↓ | AD ↑ | EAR ↑ | AD ↑ | EAR ↑ | AD ↑ | EAR ↑ |
| A) Raw Data Sharing [4] | 6 MB | 0.7619 | 0.7126 | 0.3522 | 0.6440 | 0.3165 | 0.8650 |
| B) GAT [37] | 0.6KB | 0.7588 | 0.6565 | 0.2272 | 0.6041 | 0.2960 | 0.7505 |
| C) Collab-GAT [38] | 0.6 KB | 0.5983 | 0.7020 | 0.2561 | 0.6204 | 0.5018 | 0.7195 |
| D) Compressed Feature Sharing [6] | 510 KB | 0.7066 | 0.7485 | 0.4962 | 0.6603 | 0.4690 | **0.8651** |
| E) Ours w.o. Edge Attribute | 4.8 KB | 0.7434 | 0.7322 | 0.4148 | 0.7427 | 0.2564 | 0.7785 |
| F) Ours (All Elements) | 4.9 KB | **0.9265** | **0.8336** | **0.6070** | **0.7670** | **0.6451** | 0.7846 |
| Improvements (F/D) | 104.08 | 1.3224 | 1.1137 | 1.2233 | 1.1616 | 1.3755 | 0.9069 |

lanes across from it, including any vehicles that might be proceeding straight.

- **Street Crossing**: While the ego car is crossing the street, another vehicle runs the red light. The sensing system is unable to detect this vehicle due to the presence of lined-up vehicles waiting to make a left turn.

For each scenario, we collect 24 trials, of which 12 trials are used for training and 12 trials are used for testing. Each trial contains 300 data instances and each data instance includes RGBD images observed by an arbitrary number of connected vehicles, the GNSS positions and orientations of vehicles, the command of ego vehicle, and the ground truth of vehicle actions. The data collection frequency is 10 Hz.

We use a sequence of 15 frames for each spatiotemporal graph generation. we use YOLOv5 [41] to detect objects and use SORT [42] to track objects. We extract the position as each node's attribute, which is obtained from the depth images. The spatial edges of non-ego objects are fully connected and spatial edge attributes (distances) are calculated from pairs of objects' positions. The temporal edges are connected via the object tracking results and temporal edge attributes (time gap) are identified via the temporal tracking and timestamps. The GNSS positions and orientations of each vehicle are represented as a transformation matrix with respect to the world coordinate.

In the implementation of our network, the heterogeneous attentional graph network $\phi$ is implemented based on the PyTorch geometric library. We set the number of layers of the network to be 2 and set $\mathbf{W}_v \in \mathbb{R}^{4\times 12}$, $\mathbf{W} \in \mathbb{R}^{12\times 6}$ $\mathbf{W}_e \in \mathbb{R}^{1\times 6}$ and $\mathbf{W}_b \in \mathbb{R}^{6\times 6}$ separately. In addition, we set $\mathbf{q} \in \mathbb{R}^{6\times 1}$ defined in Eq. (3) and the multi-head number is 4. The MLP defined in Eq. (8) consists of three linear layers, the first two layers. In all the experiments, we use ADAM [43] as the optimization method. We run 100 epochs to train our approach.

For the comparison, we first implement a baseline method that is **our full approach but without considering edge attributes**, including object distances in spatial edges and time gaps in temporal edges. In addition, we compare our approach with four existing methods, including

- **Raw Data Sharing** that directly integrates raw Lidar observations and utilizes the Point Transformer to extract features for collaborative decision-making [4]. As direct processing raw data on standard computers is unfeasible, we downsample the raw Lidar data size to 4096.
- **GAT** that uses graph representation generated from single-vehicle observations and graph attention neural network to encode the representation for connected autonomous driving [37].
- **Collab-GAT** that is similar to GAT but uses multi-vehicle observations as input for collaborative decision-making in connected autonomous driving [38].
- **Compressed Feature Sharing** that extracts compressed features from multi-vehicle observations and integrates the compressed features via voxel pooling to encode the situational awareness for collaborative decision-making.

None of these comparison methods can utilize temporal information due to the communication bandwidth constraint and the model design.

We employ three metrics for the evaluation of connected autonomous driving, including

- Package Size (**PS**) is the size of the shared package between connected vehicles, which is used to evaluate communication efficiency.
- Accident Detection (**AD**) is defined as the ratio of detected accident-prone cases overall ground truth accident-prone cases. Given the ground truth control of the ego vehicle, the accident-prone case is when the ego vehicle takes brake actions.
- Expert Action Rate (**EAR**) that is defined as the ratio of correct reproduced expert actions over all the number of expert actions, which is used to evaluate the imitation performance.

### *B. Results on Connected Autonomous Driving*

The CAD simulation contains a variety of challenges to perform connected autonomous driving, including complex interaction among street objects (e.g., vehicle yielding), strong occlusion in observations, limited communication bandwidth, missing objects in observation sequences, and highly dynamic situations. We run our approach on a Linux machine with an i7 16-core CPU, 16G memory, and RTX 3080 GPU. The average running speed is approximately 30 Hz. For each run, the spatiotemporal representation takes around 30 ms and the network forward process takes around 3 ms.

The quantitative results are presented in Table II. We can see that our approach performs the best on accident detection **AD** in three scenarios, which indicates the *best performance*

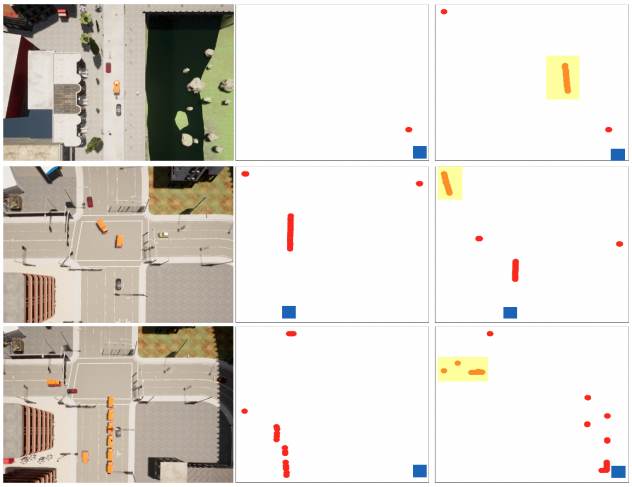

Fig. 4. Qualitative Results. The first column is the detected accident by our approach. The second column is the representations (ignore edges for simplification) of ego observation sequences, where the blue rectangle denotes the ego vehicle. The third column presents the representations of collaborative observation sequences, where the yellow areas denote the occluded objects that are not shown in the ego observations. The length of lines denotes the temporal sequence length.

*in guaranteeing safety in autonomous driving*. It is because of the capability of integrating multi-vehicle observations and temporal observations. In addition, our approach achieves **over 100 times reduction** in the shared package size. This achievement substantially alleviates the communication bandwidth burden, rendering it highly suitable for real-world applications. Furthermore, we can observe that **GAT** methods without considering temporal and collaborative observations perform worst among all comparisons. The **Raw Data Sharing** method performs better than **GAT** and **Collab-GAT**, which indicates the importance of integrating temporal cues and collaborative observations for decision-making. However, its package size for sharing is 6 MB, which is far beyond the standard 200 KB and 720 KB in DSRC and C-V2X. The **Compressed Feature Sharing** method performs better than the **Raw Data Sharing** method on both communication efficiency and driving performance. However, the compressed features still need the package size to be 510 KB for a single observation. By efficiently encoding spatiotemporal cues, our baseline model outperforms all the other methods in most cases, especially on the improvements in driving safety indicated by the higher value of accident detection. Our full approach performs the best by further explicitly encoding distances and time gaps indicated by the edge attributes.

The qualitative results are presented in Figure 4. We can clearly observe that our approach can effectively detect accidents and take brake actions. It is because our proposed spatiotemporal graph representations can effectively preserve the spatiotemporal relationships of street objects in a communication-efficient way. Furthermore, only using ego observation with strong occlusion may trigger accidents. By integrating collaborative observations, the occluded objects are correctly detected, thus the ego vehicle can make safe and effective decisions to avoid traffic accidents.

TABLE III

ABLATION ANALYSIS ON MODEL COMPONENTS: TEMPORAL COHERENCY AND DATA SHARING

| Method | Overtaking | | Left Turn | | Street Crossing | |
|---|---|---|---|---|---|---|
| | AD ↑ | EAR ↑ | AD ↑ | EAR ↑ | AD ↑ | EAR ↑ |
| NT-NS | 0.7588 | 0.6565 | 0.2272 | 0.6041 | 0.2960 | 0.7505 |
| T-NS | 0.5873 | 0.7018 | 0.2271 | 0.6564 | 0.1762 | 0.7575 |
| NT-S | 0.5983 | 0.7020 | 0.2561 | 0.6204 | 0.5018 | 0.7195 |
| Full (Ours) | **0.9265** | **0.8336** | **0.6070** | **0.7670** | **0.6451** | **0.7846** |

### C. Ablation Study

We also conduct an ablation study to analyze the components of our approach. Specifically, we test our approach in the following scenarios:

- No Temporal and No Sharing (**NT-NS**) that uses the same model as ours but only uses the ego vehicles' observation. In this case, the spatiotemporal graph is downgraded to a regular graph, thus using the traditional graph attention network to deal with it.
- No Temporal with Sharing (**NT-S**) that uses the same network as above to aggregate collaborator observations but without using observation sequences.
- Temporal but No Sharing (**T-NS**) that uses the same model as ours with temporal observations of ego vehicle and without using collaborators' observation sequences.

The ablation study results are presented in Table III. We can see that **NT-NS** performs the worst in the scenario of left turn and street crossing. By adding temporal and collaborative observations, **T-NS** and **NT-S** achieve better imitation performance (indicating by **EAR**) compared with **NT-NS**, further proving the importance of collaborative and temporal information for decision-making in connected autonomous driving. By considering both cues, our full approach achieves the best performance.

## V. CONCLUSION

We propose a novel approach for collaborative decision-making in connected autonomous driving to avoid traffic accidents. Our approach significantly reduces the sharing data size by representing a sequence of observations as a spatiotemporal graph. Then, design a novel spatiotemporal graph neural network based on heterogeneous graph learning to perform collaborative decision-making, which can analyze the spatial and temporal connection of objects in a unified way. Experimental results have shown that our approach outperforms the existing methods in communication efficiency, driving safety, and imitation performance.

Our approach has some limitations, offering possible future directions. First, we assume the transformation of collaborators' observations is based on an accurate GNSS sensor. Deploying our approach with noisy global coordination or in GPS-denied environments can be studied in the future. Second, the current spatiotemporal graphs only encode object topology information, we can further add more complex relationships into the graph representation given heterogeneous graph learning, such as adding lane information or geometric information of vehicles, to improve the performance.

## REFERENCES


[1] P. Gao, S. Siva, A. Micciche, and H. Zhang, "Collaborative scheduling with adaptation to failure for heterogeneous robot teams," in *2023 IEEE International Conference on Robotics and Automation*, 2023.

[2] P. Gao, Q. Zhu, and H. Zhang, "Uncertainty-aware correspondence identification for collaborative perception," *Autonomous Robots*, pp. 1–14, 2023.

[3] B. Reily, T. Mott, and H. Zhang, "Adaptation to team composition changes for heterogeneous multi-robot sensor coverage," in *2021 IEEE International Conference on Robotics and Automation*, 2021.

[4] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

[5] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," *Advances in Neural Information Processing Systems*, 2021.

[6] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[8] L. Gallo and J. Härri, "Short paper: A LTE-direct broadcast mechanism for periodic vehicular safety communications," in *IEEE Vehicular Networking Conference*, 2013.

[9] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, "Collaborative Perception in Autonomous Driving: Methods, Datasets and Challenges," *arXiv*, 2023.

[10] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 02, pp. 6–7, 2014.

[11] Z. Song, F. Wen, H. Zhang, and J. Li, "A cooperative perception system robust to localization errors," in *IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[12] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1852–1864, 2020.

[13] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *IEEE 39th International Conference on Distributed Computing Systems*, 2019.

[14] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2020.

[15] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *2020 IEEE International Conference on Robotics and Automation*, 2020.

[16] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," *Advances in Neural Information Processing Systems*, 2022.

[17] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "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*, 2019.

[18] J. Guo, D. Carrillo, S. Tang, Q. Chen, Q. Yang, S. Fu, X. Wang, N. Wang, and P. Palacharla, "CoFF: Cooperative spatial feature fusion for 3-d object detection on autonomous vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 078–11 087, 2021.

[19] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-vehicle communication for joint perception and prediction," in *European Conference of Computer Vision*, 2020, pp. 605–621.

[20] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno, "Multi-robot collaborative perception with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2289–2296, 2022.

[21] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2X-VIT: Vehicle-to-everything cooperative perception with vision transformer," in *European conference on computer vision*, 2022.

[22] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation*, 2022.

[23] R. Bi, J. Xiong, Y. Tian, Q. Li, and X. Liu, "Edge-cooperative privacy-preserving object detection over random point cloud shares for connected autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 979–24 990, 2022.

[24] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers," *arXiv*, 2022.

[25] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3054–3061, 2022.

[26] Y. Hu, Y. Lu, R. Xu, W. Xie, S. Chen, and Y. Wang, "Collaboration helps camera overtake LiDAR in 3D detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[27] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *European Conference on Computer Vision*, 2020.

[28] A. Meliou, A. Krause, C. Guestrin, and J. M. Hellerstein, "Nonmyopic informative path planning in spatio-temporal models," in *AAAI*, 2007.

[29] P. Gao, B. Reily, R. Guo, H. Lu, Q. Zhu, and H. Zhang, "Asynchronous collaborative localization by integrating spatiotemporal graph learning with model-based estimation," in *2022 International Conference on Robotics and Automation*, 2022.

[30] P. Gao, R. Guo, H. Lu, and H. Zhang, "Multi-view sensor fusion by integrating model-based estimation and graph learning for collaborative object localization," in *2021 IEEE International Conference on Robotics and Automation*, 2021.

[31] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[32] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, 2018.

[33] J. Chen, X. Wang, and X. Xu, "GC-LSTM: Graph convolution embedded lstm for dynamic network link prediction," *Applied Intelligence*, pp. 1–16, 2022.

[34] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv*, 2017.

[35] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, 2019.

[36] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[38] Y. Xie, Y. Zhang, M. Gong, Z. Tang, and C. Han, "Mgat: Multi-view graph attention networks," *Neural Networks*, vol. 132, pp. 180–189, 2020.

[39] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017.

[40] H. Qiu, P. Huang, N. Asavisanu, X. Liu, K. Psounis, and R. Govindan, "AutoCast: Scalable infrastructure-less cooperative perception for distributed collaborative driving," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications, and Services*, 2022.

[41] J. Glenn, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[42] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *IEEE international conference on image processing*, 2016.

[43] Z. Zhang, "Improved ADAM optimizer for deep neural networks," in *IEEE/ACM 26th International Symposium on quality of service*, 2018.