POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Vehicle-to-Vehicle Cooperative LiDAR Perception Based on Graph Neural Networks

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Vlad Marian Cimpeanu

Advisor: Prof. Matteo Matteucci

Co-advisor: Lorenzo Cazzella

Academic year: 2023-2024

1. Introduction

The imminent rise of autonomous driving, slated for realization by 2030, promises a transformative era marked by enhanced safety, comfort, and operational efficiency. The journey from driver assistance to fully autonomous systems presents challenges, with LiDAR technology playing a crucial role.

However, challenges like occlusion and limited field of view hinder seamless autonomy. Failures in overcoming these challenges can lead to dangerous situations, as observed in incidents involving bad weather and detection failures.

To address these challenges and enhance spatial awareness, vehicles can exploit Vehicle-to-vehicle (V2V) communications to exchange sensory information. This process takes the name of cooperative perception.

Nevertheless, transmitting vast LiDAR data poses practical challenges due to the impracticality of transmitting raw data with existing communication technologies. This paper explores the cooperative perception of point clouds, using a Grid-GCN model for point cloud segmentation to determine data points crucial for transmission.

The main contributions of this paper are the following:

- We present a *novel deep learning-based cooperative perception method* that, differently from the existing cooperative perception approaches, proposes to use a graph neural network to identify which points belonging to a point cloud acquired by a vehicle are worth to be transmitted to another vehicle. This approach allows the transmission of valuable raw data without overloading the network.
- We develop a *simulation framework* for testing our cooperative perception algorithm. Differently from works that have already provided a simulator based on SUMO [1] and CARLA [2], we integrate these simulators with the GEMV^2 [3] to simulate the V2V communication channel accounting for the geometry of the urban scenario.

2. Related Works

This section provides a synthetic but concise overview of the literature on cooperative perception.

2.1. Cooperative Perception Literature

The literature proposes different approaches for the cooperative perception task. Each method has its advantages and disadvantages.

In the **late collaboration** framework, each vehicle uses the data acquired through its sensors to perform object detection. Therefore, vehicles cooperate by exchanging the speed and position of the detected objects, increasing the overall perception. Even though late collaboration saves bandwidth, it is susceptible to agent positioning mistakes and experiences estimation errors and noise due to insufficient local observation.

The **early collaboration** approach aggregates raw data from every vehicle to support an integrated viewpoint. As a result, every vehicle may carry out the subsequent processing and complete perception from a holistic viewpoint, which can effectively resolve the long-range and occlusion problems that arise in single-agent perception. Arnold *et al.* [4] demonstrate that early cooperative perception fusion can recall more than 95% of objects, contrasting with the 30% for single-point perception in the most challenging scenarios. Nevertheless, exchanging raw sensory data needs extensive connection and can overload the communication network with large amounts of data, which makes it impractical for most applications.

The **Compression/decompression** approach proposes a setting where each vehicle compresses its point cloud data to adhere to bandwidth constraints. The compressed data is then broadcast to other vehicles. Upon reception, vehicles decompress the message and fuse the received point cloud with their data.

However, this system assumes all vehicles employ identical compression and decompression algorithms. This assumption poses challenges when establishing communication between vehicles with diverse equipment and cooperation models.

As a result, we investigate the possibility of identifying and transmitting only relevant points from acquired point clouds to the receiver.

2.2. Deep learning for point cloud analysis

Provided that our goal is to determine which points belonging to a point cloud are worth the transmission, we investigate the architectures that take as input point clouds. Different approaches are proposed for point cloud processing. **Voxel-based models** are a family of models that bring point clouds to spatially quantized voxel grids. These models apply 3D CNNs to the volumetric point cloud representation to leverage the spatial correlation among close regions of the point cloud. Nevertheless, to maintain the granularity of the data placement, high voxel resolution is necessary. Processing massive point clouds is expensive because of the cubic growth in computing and memory requirements with voxel resolution. Furthermore, as most point clouds have approximately 90% empty voxels [5], processing no information may waste a large amount of computing resources.

Another family of models is **Point-based models**. These models achieve permutation invariance of the input by using pooling to aggregate the point features. Point-based models define set functions to achieve order invariance. The computation cost in point-based methods grows linearly with the number of input points, making it appealing for cooperative perception purposes. However, the algorithms used to downsample the point cloud become the bottleneck, making these methods challenging to scale to large inputs.

An alternative family of models is the **Graph Neural Networks (GNN)-based models**. The point cloud is cast into a graph $G(V, E)$ with vertices V and edges E. Each point of the point cloud is a vertex of graph G, while a directed edge connects each point to all its neighbors in the geometric space. Graph-based methods are effective in segmentation tasks as they can leverage the spatial information intrinsic to the graph data structure. Moreover, these methods have a low memory footprint with respect to voxel-based models. However, as point-based methods, data structuring poses a computational bottleneck for these algorithms. Xu et al. [6] propose Grid-GCN (GGCN), which combines the memory footprint efficiency of graph-based methods and the data structuring of volumetric-based methods to increase computing efficiency. GGCN has been proven to be computationally efficient. Moreover, it shows competitive performance in segmentation tasks. Hence, we decide to employ GGCN architecture for our architectures.

3. Proposed Method

We consider a vehicular urban setting covered and served by a single Road Side Unit (RSU), i.e., a communication node deployed along a road or on the roadside providing connectivity with the infrastructure to the vehicles crossing the scenario.

Let $\mathcal{V}_t = \{1, \ldots, V_t\}$ be a set of vehicles v served by the RSU at time step t.

In our system, vehicles focus their attention on a specific area within the field of view of sensors to perform critical decision-making tasks. This area is called the Region of Interest (RoI).

Each vehicle v_{rx} is interested in pairing with another vehicle v_s to extend its environment perception by receiving information from v_s. However, the maximum amount of points that can be sent is constrained by the maximum bandwidth available.

Assuming that v_{rx} at timestep t has a position $l(t)$, speed s and heading h, we aim to determine if, given this information, v_s can learn which are the points $\mathcal{P}_{max} = \{p_1, p_2, \ldots, p_m\} \subset \mathcal{P}_s$ that maximize the satisfaction of v_{rx}, where $\mathcal{P}_s$ is the point cloud acquired by the vehicle v_s.

Given a point $p \subset \mathcal{P}_{max}$, we define the *satisfaction* of v_{rx} as:

$$\mathcal{S}(p) = \mathcal{R}_{v_{rx}}(p) \cdot \alpha(p) \cdot \eta(p), \tag{1}$$

where $\mathcal{R}_{v_{rx}}$ is the RoI score function, while α is the Age of Information (AoI) function defining the elapsed time from the acquisition of point p. Finally, $\eta(p)$ is the novelty score associated with p, and it measures how much information p adds to the v_{rx}'s point cloud $\mathcal{P}_{rx}$.

Assuming that the v_{rx} position, heading, and speed are known to vehicle v_s and that all vehicles use the same functions $\mathcal{R}$ and α, v_s can compute both $\mathcal{R}_{v_{rx}}(p)$ and $\alpha(p)$. Hence, the problem reduces to learning the function $\eta(p)$.

We model the cooperative point selection problem as a binary segmentation problem. We aim to learn a function $\mathcal{M} : \mathbb{R}^{m \times h} \cup \mathbb{R}^4 \to \{0, 1\}^m$, where m is the number of input points and h is the number of features per point. The function $\mathcal{M}$ should approximate the function u:

$$\forall p \in \mathcal{P}_s, \quad u(p, \mathcal{G}) = \begin{cases} 1 & \text{if } \eta(p) > \beta. \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathcal{G}$ is a vector containing contextual information on the position and heading of vehicle v_{rx}, while β is a tunable threshold.

In GNNs, it is common to incorporate such information in a global node. Nevertheless, GGCN does not provide this functionality. For this reason, we adapt the GGCN architecture to our needs.

Since vehicles communicate with a high frequency, the environment does not change abruptly between two consecutive time slots. In this situation, we want to avoid the sender vehicle v_s sending to v_{rx} the same data sent in the previous time steps.

For this purpose, we introduce the *Age of Transmission* (AoT) feature, which is assigned to each point of the vehicle's point cloud. The AoT of a point p is a proxy for the time elapsed from the last time the point p was sent to v_{rx}. A vehicle assigns the AoT to the point p acquired or received at time t_0 with the following:

$$\alpha_{AoT}(p, t) = \begin{cases} 0 & \text{if } t = t_0. \\ 1 & \text{if } p \text{ is sent at } t. \\ e^{-\lambda_{AoT}(t - t_0)} & \text{otherwise.} \end{cases} \tag{3}$$

Assuming that the learned model $\mathcal{M}$ captures the spatial relation between the input points, we also aim the function M to learn that points having low AoT close to points with high AoT should not be sent because the receiver already has enough information about that specific region.

4. Results

4.1. Simulation setup

To perform our experiments, we implement a new simulator providing sensor and communication data. Our simulator relies on the SUMO [1] simulator for the microscopic traffic simulation, while we employ the CARLA [2] simulator to simulate the LiDAR acquisitions. Finally, we exploit GEMV^2 [3] to compute comprehensive information regarding the communication channels between vehicles. All the mentioned simulators are open source.

4.2. Training procedure

The problem formulated in Section 3 can be split into two complementary subtasks:

- Spatial task: the aim is to learn the relationship between the position of the sender vehicle, the receiver vehicle's position, and the sender's acquired points. In other words, we are interested in learning which, among the sensed points by the sender, are the points the receiver is spatially interested in.
- Temporal task: the goal is to learn the influence of points with high AoT on close points with low AoT. Thus, the sender vehicle should learn to avoid sending points belonging to the same spatial regions for consecutive steps.

In the following, we will refer to the whole training task as joint task. We split the model training into two distinct training phases: the first phase involves performing offline training to learn the spatial task; during the second phase, we perform online fine-tuning to learn the joint task.

The reasons behind this decision are multiple. Firstly, solving the main task can be more challenging than its subtasks, causing slow convergence during training.

First, learning to solve the main task can be harder than its subtasks. Thus, the model can slowly converge when trained to solve the former. Furthermore, learning the main task can be time-consuming because the model must be trained online. In online training, the simulator introduces a nonnegligible overhead, which slows down the training procedure—a simulation step can require up to 4 times the time needed for our model forward pass. However, learning the spatial task does not require online training, as it does not demand a dataset capturing the temporal correlation between two consecutive timesteps of a communication.

4.3. Experimental results

To illustrate the significance of learning the joint task as opposed to just the spatial task, we compare the Transfer Learning (TL) model (trained only on the spatial task) against the model fine-tuned on the joint task. Fig. 1 shows that both models can learn the spatial task. Indeed, during the first communication step, the accuracy is around 83%. In contrast, the fine-tuned model outperforms the TL model in the next communication steps, which means the FT model learns to avoid sending redundant data. Figure 2 confirms the accuracy drop is induced by the fact the number of redundant points sent increases with the simulation steps for the TL model. Therefore, the introduction of the Age of Information shows an improvement in the overall performance of the model.

We notice that the primary challenges involve the speed of convergence and the time required for training. Training the model using a Nvidia Quadro RTX 6000 GPU takes approximately one hour per epoch for offline training and up to four hours per epoch for online training. The difference in computation time between the online and offline train highlights the heavy overhead introduced by the simulator. Computational constraints prevent exhaustive research over hyperparameters to be performed by means of heuristic approaches.

5. Conclusions

In this paper, we introduced a cooperative perception method wherein connected vehicles effectively choose LiDAR points to transmit, mitigating network overload. Our approach involves learning the points that the receiving vehicle is interested in but cannot perceive due to occlusions. Additionally, we proposed the concept of the Age of Transmission (AoT) to reduce redundant data transmission across multiple communication steps.

We developed a simulation framework based on the SUMO vehicular simulator, the CARLA automotive simulator, and the GEMV^2 V2V channel simulator to generate a realistic synchronized dataset of LiDAR acquisitions and V2V channel data.

Experimental results show that our algorithm can detect important areas that cannot be perceived by the receiver vehicle with mean 81%

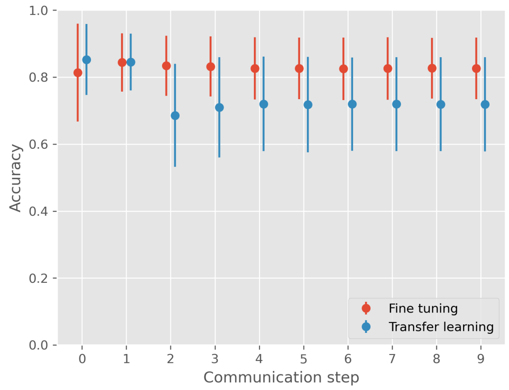

Figure 1: Error bars of Transfer Learning (TL) model compared to fine-tuned model (FT). The error bar measures the average accuracy and its standard deviation, computed for the accuracies belonging to the 95th percentile. The error bar is computed for each step of the communication. Error bars are not perfectly aligned to their corresponding step tick. This is intended to make more clear the comparison of the two models for the same tick.

validation accuracy over different communication bandwidths. Furthermore, it is shown that by introducing the AoT, data redundancy is minimized as, on average, only 20% of the available redundant points are sent.

While the findings are promising, certain limitations highlight areas for potential improvement. Firstly, the model is trained on data from a single simulation within the same urban scenario. Expanding the training to encompass multiple scenes is expected to enhance the model's performance. Additionally, the substantial overhead introduced by the simulator prevents on exhaustive exploration of the hyperparameters. As a prospective work, we suggest a more in-depth investigation into new hyperparameter settings and different architectures is warranted.

References

[1] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL `https://elib.dlr.de/124092/`.

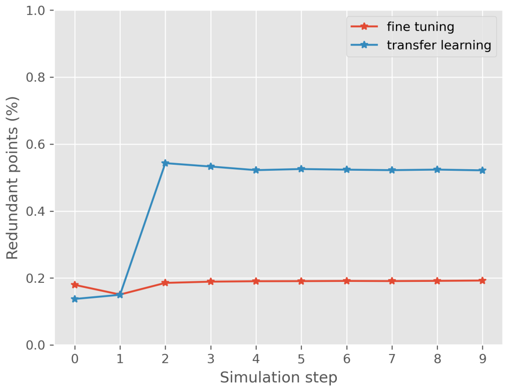

Figure 2: Percentage of redundant points sent per communication step. The percentage is computed as the ratio between redundant points sent and the total points sent.

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[3] M. Boban, J. Barros, and O. Tonguz. Geometry-based vehicle-to-vehicle channel modeling for large-scale simulation. *IEEE Transactions on Vehicular Technology*, 63 (9):4146–4164, Nov 2014. ISSN 0018-9545. doi: 10.1109/TVT.2014.2317803.

[4] Eduardo Arnold, Mehrdad Dianati, Robert de Temple, and Saber Fallah. Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1852–1864, 2020.

[5] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020.