POLITECNICO
MILANO 1863

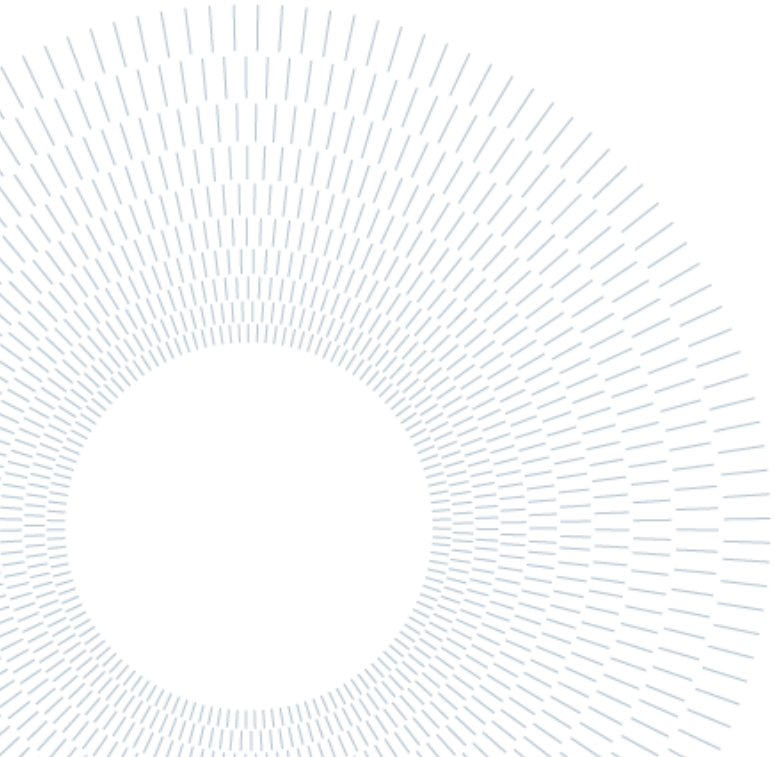
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Exploring the Different Aspects of V2V Communication Systems for Specifying the Semantic Empowering Capabilities Regarding Cooperative Perception Application

TESI DI LAUREA MAGISTRALE IN
TELECOMMUNICATON ENGINEERING
INGEGNERIA DELLE TELECOMUNICAZIONI

Author: **Ali Sourchap**

Student ID: 10772181
Advisor: Maurizio Magarini
Academic Year: 2021-22



Abstract

In recent years, thanks to machine learning flourishing, countless studies have attempted to go beyond content-agnostic message transmission and take care of messages' purpose and semantic meaning. Due to the promised efficiency and reliability advancements that would be obtained by switching to semantic communication, we decided to study V2V communication systems (more specifically, V2V communication for the "cooperative perception" application) to detect possible approaches for semantic empowering.

Within the cooperative perception application, systems mostly have two difficulties: sharing the massive captured data and managing data transmissions. Therefore, we studied this application-based communication system from two distinct aspects: semantic-empowered physical layer and network managing.

In the scale of the physical layer, our objective was to find the proper goal-oriented encoding/decoding modules that can extract and recover the semantic features. By analysing some proposed DNN learning-based pairs of encoder/decoder that address mentioned challenges, we found the great opportunity of employing these modules in the V2V systems to obtain significant compression rates with the limited accuracy declining. Furthermore, in the scale of the network managing, we proposed a novel factor graph-based planning system that predicts impending collisions by exploiting the spatial grabbed data of the traffic environment. Concerning the network condition, this system employs combinatorial optimization to pick the best massage combinations to avoid potential collisions.

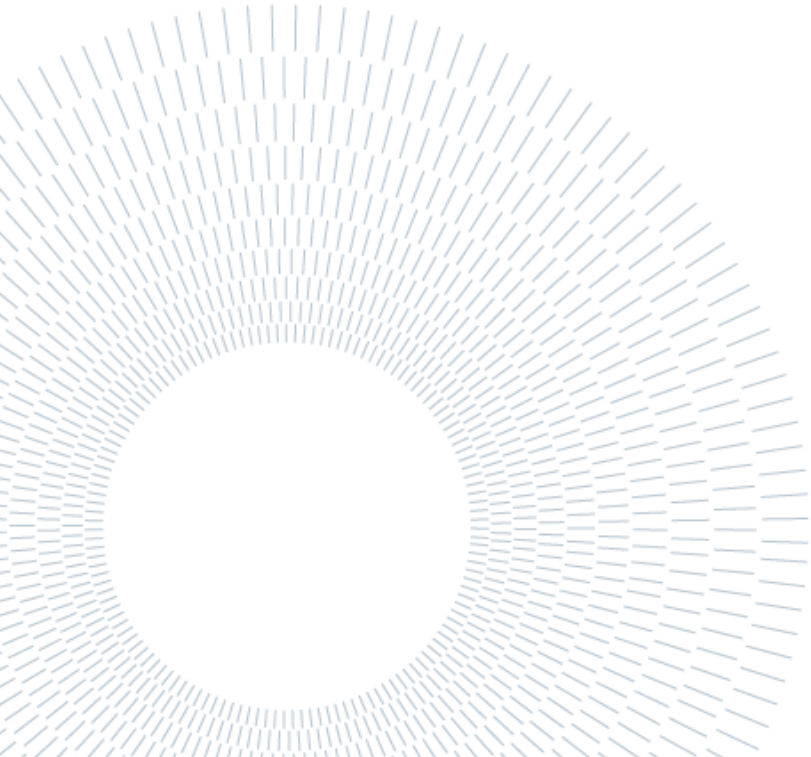
Although we designed a program for assessing our method, the complete assessment was not feasible due to the shortage of time. Nevertheless, this study reveals an existing gap between the conventional and goal-based criteria used in network management. Our proposed method does not just have the potential to be executed in real scenarios but also promotes the idea of making goal-oriented strategies. The new intelligent networks, with smart entities and more specific ultimate goals, can utilize the available data in the network to control it more efficiently with respect to communication goals. Indeed, our proposed system evidently identifies this issue and addresses it by presenting an innovative way to make a goal-oriented network.

Key-words: Semantic Communication, Goal-oriented Communication, Vehicle-to-vehicle, Cooperative Perception, Factor Graph, Belief Propagation.

Abstract in Italiano

Negli ultimi anni, grazie allo sviluppo del machine learning, numerosi studi hanno cercato di andare oltre una semplice trasmissione del messaggio al fine di occuparsi anche dell'obiettivo del messaggio e del suo significato semantico. Grazie ai miglioramenti in quanto a efficienza e affidabilità ottenuti da questo switching semantico, abbiamo deciso di studiare i sistemi con comunicazione V2V (in particolare, comunicazione V2V per l'applicazione di percezione cooperativa) al fine di rilevare possibili vie per migliorare la comunicazione tramite la semantica. All'interno dell'applicazione di percezione cooperativa, i sistemi hanno generalmente due difficoltà, cioè condividere la grande quantità di dati e gestire la trasmissione. Quindi, abbiamo studiato questo sistema di comunicazione basato su un'applicazione secondo due aspetti distinti: miglioramento semantico del livello fisico e gestione della rete. Dal punto di vista del livello fisico, il nostro obiettivo era quello di trovare i corretti moduli di encoding e decoding orientati all'obiettivo che possano estrarre e recuperare l'informazione semantica. Analizzando le coppie di encoder/decoder di tipo DNN basato sull'apprendimento che si occupano di questa sfida, abbiamo trovato l'ottima possibilità di utilizzare questi moduli nei sistemi V2V al fine di ottenere una compressione significativa limitando la perdita di accuratezza. Inoltre, per quanto riguarda la gestione della rete, abbiamo proposto un nuovo fattore basato sui grafi per il sistema di planning in grado di predire i prossimi conflitti sfruttando i dati catturati. Considerando le condizioni della rete, questo sistema usa l'ottimizzazione combinatoria per trovare la combinazione di messaggi migliori per evitare dei probabili conflitti. Siamo riusciti a scrivere un programma per provare il nostro metodo; tuttavia, il lavoro finale non è ancora completo per mancanza di tempo. Nonostante tutto, questo studio mostra un'evidente differenza tra il metodo convenzionale e quello basato sull'obiettivo per gestire la condivisione di messaggi in una rete. Il metodo proposto da noi non ha soltanto la possibilità di esecuzione in uno scenario reale, ma cerca anche di promuovere l'idea di considerare gli obiettivi durante la comunicazione. Le nuove reti intelligenti, con eventualmente degli obiettivi specifici, possono gestire in modo più efficiente le risorse della rete

Parole chiave: Comunicazioni Semantiche, Comunicazioni Mirate, Veicolo-Veicolo, Percezione Cooperativa, Grafo Pesato, Propagazione delle Credenze



Contents

List of Abbreviations

Abbreviation	Definition
BEV	Bird Eye View
BN	Batch Normalization
BP	Belief Propagation
BSM	Basic Safety Messages
CBH	Carnap and Bar-Hiller Theory
CBR	Channel Busy Ratio
CIT	Classical Information Theory
CNN	Convolutional Neural Network
CPM	Collective Perception Message
CVT	Cooperative Vehicle Training
DRL	Deep Reinforcement Learning
E2E	End to End
FEC	Feature Extraction Component
FoV	Field of View
GBP	Gaussian Belief Propagation
GNN	Graph Neural Network
GRU	Gated Recurrent Units
HSM	Hypothesis Sharing Method
IoU	Intersection Over Union
MAP	Maximum a Posteriori Estimation
MRF	Markov Random Field
NLP	Natural Language Processing
NMS	Non-maximum Suppression
ODM	Object Detection Module

PoC	Perceived Object Containers
ReLu	Rectified Linear Unit
RIS	Raw Information Sharing
RoI	Region of Interest
RPN	Region Propose Network
SDF	Signed Distance Function
SFF	Spatial Feature Fusion
SIC	Sensor Information Containers
SIT	Semantic Information Theory
SLAM	Simultaneous Localization and Mapping
SVT	Single-vehicle Training
TMA	Translation MOD-alignment
V2V	Vehicle-to-vehicle
VFF	Voxel Feature Fusion

Introduction

1.1. Semantic Communication; Next Paradigm

The classical information theory founded by Shannon, as he claimed, is oriented around symbols communication accuracy, in a way that "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point [1]." Although this deliberated focus on the technical aspect of communication was an intelligent move (that also ended up deriving a mathematical information theory based on probabilistic models), it seems it could be not obligatory anymore to keep the information transferring just limited to the classic objective of exact symbols transmission (and consequently limited to the Classical Information Theory (CIT)). Thanks to the achievements in AI and the new computing and learning capabilities, it is nearly feasible to shift the resolution of the current communications from the technical problem (how accurately can the symbols be transmitted?) to the semantic problem (how accurately can the desired meaning be conveyed?) [2].

This pristine idea of going beyond the deliberated "content-agnostic" communication paradigm to a modern one that also takes care of semantic attributes of underlying data is essential to be researched and investigated because the main expected achievements that would be gained by semantic communication would be the vital milestones towards the next possible evolutions in wireless communication. We could briefly state these advantages:

1. **Increasing Communication Reliability and Sustainability:**
   As explained in more detail in the following sections, semantic communication is not based anymore on no-error bits transmission, and even with some existing faults in the bit-scale, it would still be possible to convey the desired semantic content. that is, it would be possible to experience higher reliability in these systems. Moreover, the AI-empowered entities that inevitably would be a major part of semantic (or goal-oriented) communication systems expect to learn the ability to fill the gap of the missed information based on the other correctly received slots. It looks feasible to imagine some intelligent entities in the coming systems that can implant the missing transmitted slots with the best insertion based on their own knowledge bases and the primary communication goal.
2. **Exploiting More from the Channel Resources (in Terms of Higher Semantic Capacity and Increased "Meaningful Data" Per "Transmitted Symbol" Ratio):**
   Although the deficiency of the Classic SIT is fully put forward in its corresponding session, and we are knowledgeable about them, some analysing

tried to employ this flawed CIT to show the possible capacity rising when we switch to semantic communication. As [3] shows, for a simplified communication scenario, by using a semantic encoder with low final semantic ambiguity along with a semantic decoder that has robust interpretation ability, we might reach high-rate semantic communication while using a low-rate engineering channel, that is a tremendous outcome and a sensible reason to attempt to shift the paradigm from CIT to SIT. In addition, for goal-oriented communication, we know that the objective is to modify and adjust the transmission to reach the intended goal. One of the essential tasks to fulfilling this objective is to avoid sending unnecessary information. Therefore, we expect to experience higher "meaningful data" per "transmitted symbol" ratio in these systems as well, which means more profitable channel usage.

Since the first days of the semantic information concept, different researchers have continuously refined and modified it. At this moment, thanks to the recent AI flourishing, we have these new opportunities to surpass the early defined frameworks for Semantic Information Theory (SIT) (which mostly were modified versions of Shannon's theory [4] [5] [3]) and design the new AI-based semantic communication systems. In general, based on [6] recommendation, we can categorize the field of semantics into two general research directions: "Semantic information" and "Semantic communications." Besides them, "goal-oriented communication" could be an important research direction motivating the need to move beyond CIT. However, it is not always easy to draw a particular line to distinguish between goal-oriented and semantic communications because the amount of semantic information in a particular message must be measured with respect to the message efficacy in reaching the main goal of data exchanging [7].

For more detailed analysis, these general fields can be divided into sub-fields. These categorizing is stated initially in the compressive analysis [6]:

- **Semantic Information:**
    - **Classic Semantic Information Theory:**
      Historically, from 1953, with the Carnap and Bar-Hiller works (CBH Theory [4]), which try to modify Shannon's classic formulas to quantize the amount of semantic information, we have gained a vision of the differences between statistics and semantics attributes that exist in messages. Up to this point, there are no flawless and firm frameworks that can scientifically study the quantification of semantic information for different data types. While we cannot neglect the improvements gained in these years by some significant studies like [8], having a classical framework to study semantic information still is an open question.

- **Modern Semantic Information Theory:**
  Over these years, with recognizing the advantages that would come with applying AI Technologies as well as the interdisciplinary studies that have been started to analyze the semantic information out of the "statistical analysis" box, we have seen significant improvements around the concept of semantic information theory which has gone beyond the CBH framework. For example, [9] proposed a semantic information theory by introducing the trinity of syntactic, semantic, and pragmatic information and proved that semantic information is the unique representative of the trinity. [10] defines the amount of semantic information as the amount of syntactic information preserved by the optimal intervention that intervened in the system and environment jointly distribution. Furthermore, For a communication case, [7] gives a three-scales definition of semantic information for a communication system. Respectively, these microscopic, mesoscopic, and macroscopic scales advocate for assessing and extracting the semantic value of data at the source, link, and system levels. Eventually, These modern theory frameworks can offer a comprehensive viewpoint about semantics to make communication systems more potent and secure.

- **Semantic Communication:**

  We can use the human communication phenomenon as an intuitive explanation for semantic communication. In human contact, in conveying a concept from one person to others, the relevant aspect is what is communicated (the semantic content) not how the message is brought to the destination (symbol reproduction). With the same insight, correct semantic transmission happens when the concept of the transmitted message is interpreted correctly by the destination side, which does not necessarily imply the whole bits no-error decoding. One of the main reasons the semantic level offers a significant performance improvement is because of using a shared knowledge base between both sides. The advantage of having this knowledge base (which in general level consists of entities and a set of logical rules) at the receiver side is giving the chance to correct errors occurring at the level of the bits and symbols. There are also other research directions around this latest communication system. In the following, some of these main branches are explained:

  - **Semantic-empowered Physical-layer Transmission:**
    As we stated earlier, the main goal of semantic communication is not ensuring accurate symbol reception at the destination anymore, and the main goal has changed to accurately recover the semantic information at the destination. One main advantage of this new goal is improving the physical layer transmission efficiency. The reason for this improvement

is the latest coding methods features that can be considered since the purpose of this transmission is no longer bounded to exact symbol reproduction, and the more general goal of conveying the meaning of data can compensate for some possible faults in bit level. Usually, this field of exploiting the mentioned features is studied as semantic encoding and decoding. Since semantic communication still lacks a firm and unified framework and formulation, existing projects mostly realized these semantic encoding and decoding modules with machine learning methods. The available methods are then classified into the dis-joint design and joint design.

In dis-joint design, semantic encoder and decoder are treated as a block-wise segmentation that would add to a conventional communication system (before/after channel decoders/encoders) to reduce the decoding overhead. On the other hand, the joint design takes the semantic-enhanced joint source-channel coding route. That is, joint source-channel coding modules optimized with a common objective. We can find multiple E2E semantic architectures that combine semantics and physical layer modules to lessen semantic errors. Most of these architectures are executed by applying neural networks in the physical layer instead of the conventional source/channel coding blocks.

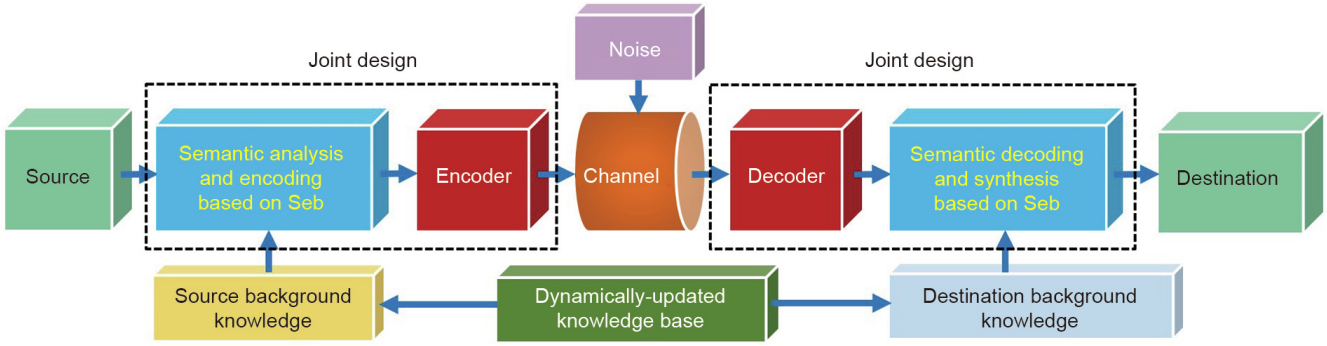


Figure 0.1 Illustration of semantic communication in the semantic-empowered physical-bearing layer [6]

- **Application-aware Communication Protocols:**

  The conventional protocols of lower-layer communication are designed concerning various upper-layer applications. However, This generality causes redundant functions that are not utilitarian for all applications. One of the systematic approaches to increasing efficiency is the cross-layer protocol design. For instance, Some existing projects have tried to propose efficient routing protocols in which the information flows in lower layers are integrated to bring the E2E delay down [11]. Moreover, [12] proposes an application-aware and cross-layer protocol that reduces the transmitted redundancy by executing a semantic-filtering mechanism. In terms of multi-agent communications, we can find some

executed projects, like [13], which try to apply some human-like communication strategy for intelligent communication agent interaction. This strategy eventually can lower resource consumption and improve efficiency. The packet forwarding framework proposed in [11] is plotted in Figure 0.2 as a cross-layer information sharing protocol. This protocol is presented for Inter-Vehicle communication systems and utilizes mobility prediction information to jointly optimize routing, MAC, and beam control of directional antennas.

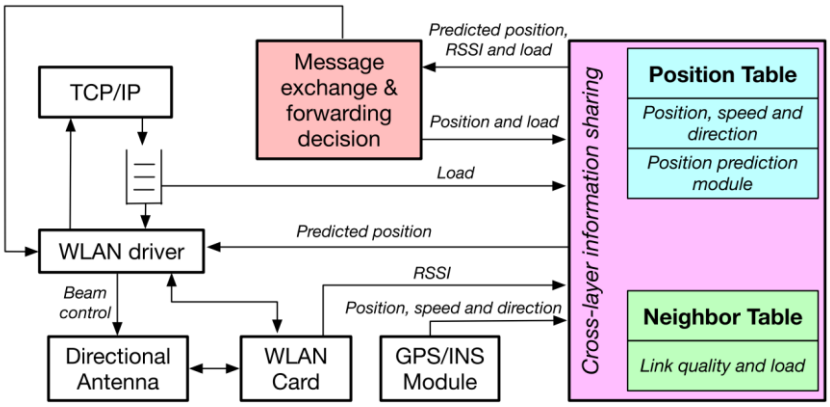


Figure 0.2 MP2R packet forwarding framework [11] as an example of application-aware communication protocols

- **Goal-oriented Communication:**
  We know that in most communication cases among interacting entities, the point of this communication among the entities is to accomplish a joint and common goal. To a greater extent, the fundamental system specification is associated with pursuing the goal. Since Successful goal achievement depends entirely on reaching the goal within the given time and resource constraints, it sounds reasonable that the communication system should be designed for the goal-related specifications and limitations. In better communication literature, this means that (for example) unnecessary information not associated with the main communication goal can be omitted because they are considered redundant data. When discussing being essential or redundant, we implicitly argue about a new level (the effectiveness level) responsible for the efficient management of goal-oriented communications. This level of semantic attribute extraction will also act by properly orchestrating the resources available at the technical level. (network nodes, the computation resources, and control)

Although the mentioned categorizing that is offered by [6] has clearly classified the different sides of semantic information and communication, it (as same as most other relevant studies) mostly emphasizes a part of semantic communication which relates to point-to-point communication and how the different entities of a general network

can modify the quality of their communication by considering the semantic of their messages. While the meaning of semantic communication could go beyond this individuate communications. Apart from the general networks whose elements likely have no relation to each other and only seek their communication objectives, we also could imagine more specialized networks of (possibly intelligent) entities that their combinations are being done in order to guarantee to reach a common goal that could be reached by network's members jointly cooperation. The entities in this kind of network (most are in industrial and advanced areas) are gathered and oriented around a common purpose that would act like a conductor to direct the performance of each entity.

A recent study that effectively addresses semantic aware networking is [7]. As we mentioned earlier, this practical study claims that the amount of semantics information in a particular message must be measured with respect to the message efficacy in reaching the main goal of the communication. Based on this fundamental assumption, this study presents the following essential properties of a semantic-empowered network for having consistent and reliable communication:

a. **Semantic Filtering:**
   Which is semantic-aware data selecting (and censoring) to avoid redundant data transmitting and only let the useful and relevant information (based on the communication determined goal) use the available communication resources.
b. **Semantic Preprocessing:**
   Which points out to entities capability for processing and doing the required computations on the available data to change, reshape or squeeze them with respect to the determined objectives.
c. **Semantic Reception:**
   Because of implemented preprocessing modules on the transmitting side, there is a need for goal-dependent data recovery and reconstruction at the reception. A semantic quality indicator could measure this segment efficiency for the recovered data.
d. **Semantic Control:**
   This functionality might be the key property in identifying semantic-empowered networks, and it mentions the goal-oriented supervising and organizing over network resources and features. This aspect of semantic empowering relies on the scale of the network, the transmitting data, and the message anterior and posterior processing procedures.

The organizing mentioned in the semantic control session in [7] is named "the orchestration." One of the technical challenges mentioned in the orchestration is "Goal-oriented Resource Orchestration." This challenge could happen in networks dealing with multimodal data. Usually, for this type of data, the network needs to pick different data gathering policies and strategies with respect to the available resources and the application requirements. The word "orchestration" could clearly show the

dynamic aspect of networks and the continuous necessity of having a proper online strategy to make the best network application-based decisions. A decision making process that [7] explains it as: "which piece of information, from which sources, and toward which destinations, and at what times, should be gathered and transmitted in order to fulfill communication constraints."

As explained in the next sessions (1,2), an intelligent network of connected and related vehicles could have a good potential for being studied from a semantic-empowering point of view. The dynamic nature of this network and the vast types of different data transmitted in this network make them good study cases that surely would have several flawed and imperfect sides, which could be improved by applying and proposing semantic communication theories.

1.2. V2V Communications Systems

Vehicle-to-Vehicle Communications Systems, known as V2V communication systems, are wireless communication systems specifically designed for vehicles to communicate in the traffic environment directly. The shared data in this communication could vary from simple reporting or signaling data (such as the data about the speed and position of vehicles or Traffic signals) to the raw sensor data (that is gathered by the different sensors of a vehicle and broadcasted for other present vehicles in the scene). As it is clear, V2V communication contains a large amount of multimodal data that must be transmitted between different vehicles. Furthermore, to find a good sense of the coming benefits of V2V communication, some of the expected advantages are explained in detail in the survey [14] stated below.

These benefits are categorized as:

- **Improving Traffic Management:**
  V2V communication systems enhance traffic monitoring and management during congestion. For instance, one communication-based traffic management is adapting to the traffic light scheduling to reduce average delay and therefore travel time. Moreover, there are more gains with respect to precision, positioning, and safety parameters obtained by V2V communication in traffic flows with high transmission efficiency.
- **Benefits Related to V2V Interoperability:**
  V2V interoperability communication makes the network of existing vehicles, which can provide enhanced mobility, reliability, and routing protocol for these vehicles. Direction providing and route optimizing are two examples of network routing protocols that the interoperability data exchanging could gain. One of the significant aspects of this V2V interoperability (we focused on it in this research) is V2V sensor data exchanging that results in enlarging vehicle sensing range and improving driving functions by expanding information sharing among vehicles. Doing away with GPS restriction in localization is

another notable advantage of V2V interoperability that can improve precision by using data fusion techniques to estimate vehicle location.

- **Benefits Related to Safety:**
  V2V communication can provide safety systems that notify drivers about the environment and path conditions. These safety notifying systems would then reduce the number of possible collisions. V2V communication systems allow vehicles to share state information to improve driving safety and network transportation efficiency. In a platoon scenario, the main objective of V2V communication is to ensure vehicles have a safe distance to avoid chain-reaction car accidents. This preservation task is being done by continuously tracking the vehicles' dynamic properties (velocity and acceleration).

Despite the Notifying messages that will bring these benefits, the other real-time data sources that are being transmitted between vehicles are:

- Positioning and Moving Related Data:
  - Vehicle Speed
  - Distance between Vehicles
  - Vehicles Direction
  - GPS/GNSS
- Vehicle main device (or sensor) data:
  - LiDAR
  - Camera
  - Radar
  - Intermediate Processed Data

Among These data, the data from the LiDAR and Camera sensors take the notable portion of Real-time data transmitted among vehicles.

1.2.1. Collaborative Perception, As a V2V Application

Mainly, this transmission aims to improve road safety and mobility criteria. One of the main goals of this sensor's data transmission could be related to collaborative perception of a dynamic environment where the vehicles pass through. The extended sensing that comes with this collaboration is critical for robust autonomous (or intelligent) driving. However, designing a reliable, collaborative perception system requires addressing the main challenge caused by limited network resources. Moreover, the inevitable discrepancies that exist for a set of different sensors (such as different sensitivity or noise levels, different built-in alignments, and set-up for each sensor) also challenge selecting and pre-transmitting processing methods needed for this data. Needless to say, the expanded perception of the environment not only has some safety benefits but also enables the conjunction among the vehicles to enhance their routing protocol, optimize the reliability of their communication, and lower the latency between communication vehicles.

In order to highlight the importance and benefits of collaborative perception, we can highlight the explanations and quantified results of [15] that analyzes the performance improvement earned by vehicular networking and collaborative sensing. The different criteria for this evaluation are coverage, reliability, and penetration. It summarizes this improvement in these numbers: Collaborative sensing significantly improves coverage from 20% to 80% with a 20% penetration.

1.2.1.1. Collaborative Perception, as the Main Goal for Executing Semantic-empowered V2V network

As we can explain, in a dynamic environment of mobile, immobile vehicles and different obstacles surrounding them, the V2V data transmission expands the individual and collective vehicles' perception of their environment. Vehicles could exploit this expanded viewing in different benefits, including V2V interoperability benefits or the benefits related to safety attributes. Therefore, treating these transmissions as opaque data carrying that does not consider the context and topic of information seem an inefficient and unreasonable way of data transfer. In this thesis, we tried to look at these types of connectivity. We proposed the ideas that are trying to modify these communication networks (which consist of intelligent elements) by taking into account the semantic attributes of the conveyed messages and the specific goal behind these messages. Passing through the classic communication framework to semantic ones needs to change our perception of the communication per se. We should not look at communication as an end, but we should look at it as a means to facilitate achieving defined goals. Based on this belief, the Semantics meaning in the context of communication networks is a measure of the usefulness of messages concerning the goal of data exchange.

For the rest of the thesis, we would mainly focus on two aspects of V2V communication systems that seem to have advancement potential with switching to the semantic paradigm with respect to cooperative perception as the central application in vehicles' data exchanging.

1.1.1.1.1. Semantic-empowered Physical Layer for V2V Systems

The first aspect that we would analyze is related to the physical layer of the V2V communications system. As we discussed, one of the cooperative perception's critical features is exchanging the captured sensor data, which sometimes could be massive and resource-demanding. Evidently, implementing some semantic-based feature extraction or data encoders in the physical layer can remarkably decrease the required communication resources for sensor data exchanging. We will explore this issue in more detail in the rest of the thesis. We are seeking possible modifications that could be applied (or even have been applied) to this layer of the system to carry semantic information related to cooperative perception. We will also point out their existing challenges and their proposed solutions.

1.1.1.1.2. Semantic-empowered Intelligent Network Managing for V2V Systems

The second aspect that could even have more room for improvement is proposing goal-based intent-driven strategies and frameworks to manage the data transmission in V2V systems. The continuous data gathering done by vehicles from the dynamic traffic scenes, along with the constant need for sharing the critical part of this vast amount of data, creates a critical need for firm network strategies. Obviously, for these kinds of busy and loaded networks, not having a solid strategy could simply end up overloaded networks filled with redundant or unnecessary data, which will stop sharing valuable.

One of the main reasons for claiming that developing a semantic-based strategy for these systems is a more reachable target is based on two reasons. The first reason is related to the absence of firm theoretical frameworks for studying the amount of semantic information in different multimodal data and proposing semantic encoder/decoders supported by a solid theoretical background. As we will see, most of the proposed semantic encoding\decoding systems are based on the statistical patterns extracted by deep neural networks.

1 Semantic-empowered Physical Layer

As we discussed in the introduction, There is a lack of a firm probabilistic logical framework to evaluate the amount of semantic information, especially for the multimodal data scenarios (which include vehicular communication cases). Moreover, when this multimodal information comes with high dimensions and great uncertainty, it would be more challenging to dig it for a proper analyzing framework. The fact that the theoretical aspect of semantic communication spends its infant stage leads to prevalent methods for semantic feature extractions based on deep learning-based tools.

One of the earliest studies that proposed using deep learning-based tools for encoding the semantic information of input is [16]. Although the processed data in this study is discrete (comes from the natural language processing (NLP) field), the proposed end-to-end (E2E) semantic communication framework, which integrates the semantic inference and physical layer communication problems, is a popular preliminary idea that could be found in a lot of other studies [17] [18] [19] [20] [21]. Over and above that, These E2E idea is proposed in mentioned survey article [6] as one of the leading scenarios for semantic extracting that could be done in the physical layer of communication systems.

For multimodal data, to achieve the goal of semantic communication, which is delivering meaning, there is a need to set up a "semantic extraction" module in the physical layer for context-based semantic encoding and decoding. Since in V2V communication, there is a wide variety of data, there is the need for multiple deep neural networks to extract semantic features of these different types of data. State of the art for machine learning-based semantic encoding/decoding as [6] categorized can be analyzed at three different levels:

1. **Semantic Encoding and Decoding:**
   According to this method, the encoding/decoding process is realized as an additional segment of the communication system, independent of other modules, such as channel coding.
2. **Joint Semantic-aware Source-channel Encoding and Decoding:**
   An integrated system jointly designed for source and channel encoding/decoding can perform semantic encoding/decoding as well.
3. **Semantic Utilization of Channel Information:**
   In order to facilitate semantic information transmission, the channel state information, like SNR, fading, and present interferences, are extracted and integrated by the semantic system.

In V2V communications, the first level of semantic encoding/decoding suits better for fulfilling the desired goal-oriented communications. Because as it is mentioned, there is a wide variety of different types of information that must be treated on the microscopic scale of semantic systems.

1.1. Deep Learning-based V2V Features Extraction

While exploring different articles about cooperative perception, precisely the data transmission issue of this application, we coincidently found fascinating studies that have composed and designed semantic preprocessing and semantic receptions modules without mentioning it! Most of these studies have addressed the high resources needed to share the raw data (captured by vision sensors) as one of the main problems in cooperative perception systems. Their solutions for this problem were deep learning-based feature extractors that could solve this problem by projecting this raw data to lower dimensions. Since the next level after receiving sensor data is the "sensor fusion" operation in the destined vehicle, these studies also proposed a decoding stage at the receiver to recover the transmitted data to use them in the sensor fusion level.

Based on the determined goal, sensor fusion and perception expansion, and by using the deep neural networks on both sides, these studies have proposed some semantic-based modules that are being applied in the physical layer of communication systems. Due to the earlier description, these studies have inattentively proposed semantic-based solutions for this challenge. They presented a good framework for studying this problem and got good results. Therefore, for the rest of this session, we will deliberately review some leading and influential ones. The kind of problems that they faced during these researches and the proposed solutions could make us more familiar with semantic-based modification. Moreover, the style of their suggested solution can be utilized in other similar problems as well.

1.2. Three Different Levels of Information Aggregation in Cooperative Perception Systems

One of the primary types of data commonly transmitted for sensor fusion purposes is (raw or processed) LiDAR data. The [22] categorizes cooperative perception methods by the type of LiDAR shared data. This type of categorizing addresses the challenges such as scalability issues (that occur because of bounded bandwidth) and inconsistency of shared information. Each of these categories is associated with a specific bandwidth consumption range and cooperative performance. The defined categories are as follows:

1. Raw information sharing (RIS).
2. Partially processed information sharing (DFS).
3. Fully processed information sharing (FIS).

Respectively, from the first to the third category, there is a trade-off between the amount of required bandwidth and final perception performance. Raw information sharing (RIS) requires more allocated bandwidth, leading to better insight expanding performance. On the other hand, fully processed information sharing (FIS) has a significantly lower communication cost. However, more detection challenges exist, such as partially observed objects and the lack of consensus between cooperative entities.

The most curious level of information sharing concerning semantic communication is the intermediate one, named Partially processed information sharing. As [22] claims, by utilizing Deep learning tools, it would be possible to find some middle grounds between the rich information content of RIS and the low communication consumption of FIS to extract, transmit, and integrate partially processed data that have been obtained by using intermediate layers of a neural network which have been trained due to the main goal of the transmission which is aggregating cooperative information with less data missing or errancy and most efficient bandwidth consumption.

The following Figure 1.1 demonstrates an overview of these three different levels of information aggregation for the general case of transmitting volumetric sensor data.

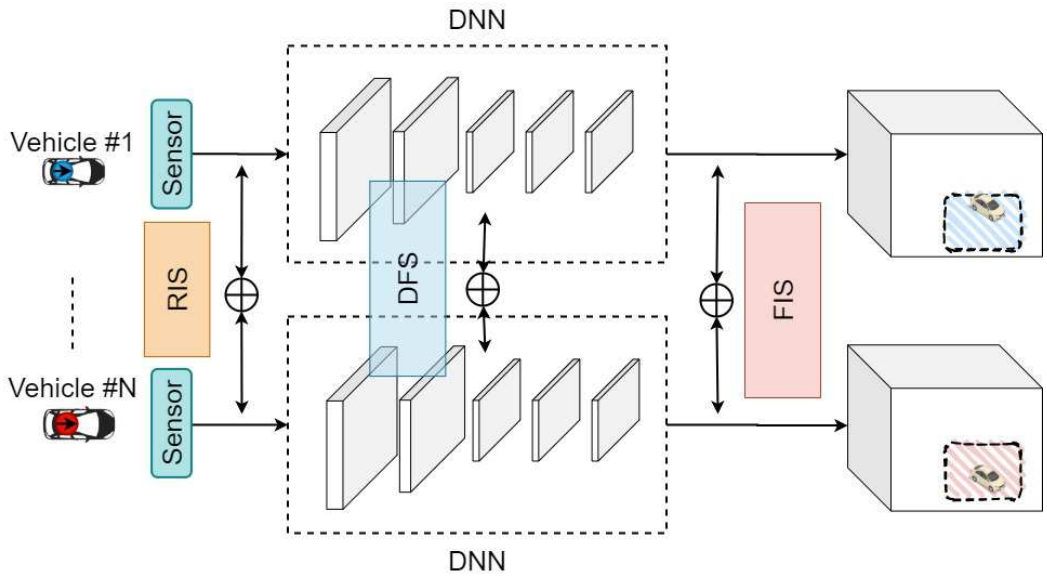


Figure 1.1 Overview of cooperative information aggregation methods [22]

One of the main fields of cooperative perception is using other vehicles' LiDAR observations. This branch is mainly studied alongside the point-cloud object detection method, which is the conventional way to utilize LiDAR data. For instance, [23] proposes cloud-based 3D object detectors designed to work on a diversity of aligned point clouds to fuse the sensor data collected from different vehicles with different

alignments. The proposed method is Sparse Point-cloud Object Detection (SPOD) which detects objects in low-density point clouds data. The architecture of the SPOD method is based on a Sparse convolution neural network that uses a single E2E DNN to operate on the raw point cloud. Other studies try to implement this cooperative perception by sharing the fully processed data in the form of some map-sharing. [24] is one of these efforts to share the extended 3D maps generated by LiDAR data to improve tracking performance in a V2V network. [25] also deals with this fully processed data sharing by considering the bandwidth restrictions and proposing an adaptive communication strategy for treating this matter.

After pointing out these studies around FIS observation exchanging, it should be noted again that although these studies considered and addressed the challenges of efficient resource and bandwidth allocation, there is no efficient addressing to the beforementioned challenges of gathered 3d maps, which are partially observed objects and lack of consensus. These challenges were the motive to investigate further and design practical methods. These investigations accelerate with recognizing the capability of utilizing DNNs to develop some methods based on Deep Feature Sharing (DFS). By investigating these methods, which rely on Deep-tools, it is possible to get some idea about the goal-oriented feature extraction that could help apply these feature extraction methods as some modules at the physical layer.

[26] proposes a method called "Feature Sharing for Cooperative Object Detection" (FSCOD) which is based on multi-layer CNN that gets the LiDAR data as the input to extract, exchange, and integrate the desired features to detect the objects in the environment in a cooperative manner. The same research teams also try to use FS-COD as a preliminary baseline architecture to devise an Adaptive Feature Sharing to dissolve the bandwidth limitation problem from the architectural design and enable cooperative vehicles to share features (or maps) with variable sizes [27]. Generally, most of these cooperative perception methods can be mainly distinguished by their *alignment* and *aggregation* method. In [22], the effect of different aggregation and alignment methods is studied. Therefore, briefly reviewing this paper can show suitable and feasible DNN methods for extracting and aggregating LiDAR data and their efficiency.

1.3. Information Gathering for Cooperative Perspective Purposes

As we discussed the cooperative perception could be categorized by the type of shared data:

1. Raw Data Sharing
2. Deep Feature Sharing
3. Fully processed Sharing

These categories are associated with the different mathematical spaces. When the Raw data is being shared, the transferred information belongs to the sensory vector space. While we transferred the features extracted by deep learning tools, it belongs to an intermediate space defined to embed the sensory information and share it among vehicles. Eventually, when the final information is shared, the state space is determined by the purpose and goal of communication. In other words, these hypothesis spaces represent the semantic (innate and contextual) attributes that are interpretable. Take into account that although many fusion and cooperative algorithms have been introduced based on sharing finally processed data [28] [29] [30], the hypothesis final space might not necessarily be optimal with regards to the desired goal of this kind of V2V communication (which is joint cooperative perception). Reasoning that the hypothesis space is defined for the task of detection does not mean that it necessarily includes the optimal information for efficient volumetric sensors transmission and fusion. In other words, the mentioned space is not defined with respect to compression, which might lead to redundancies or information loss issues.

The drawback of transferring fully processed data, mentioned in the past paragraph, is no longer a problem for deep feature sharing since the intermediate space of deep features can be regulated to increase the performance of the desired goal. (that could be compression or fusion) The main challenge in sharing deep extracted features is that the semantic attributes space is not interpretable. Therefore, there might be no trivial way to perform some operations on the feature vectors—the operations like alignment that is an essential operation for sensor fusion.

At the end of the efficiency spectrum, RIS from redundancies. In many cases, such redundant data gathering is not beneficial. However, redundant information may be useful in some cases to improve performance. For example, this information can be used in noisy scenarios to increase object detection performance. Nevertheless, even in these scenarios, the cost of allocated sources (bandwidth) could be higher than the gain in slightly increasing the performance.

The following Figure 1.2 illustrates the proposed methodologies for cooperative object detection, cited from [22]:

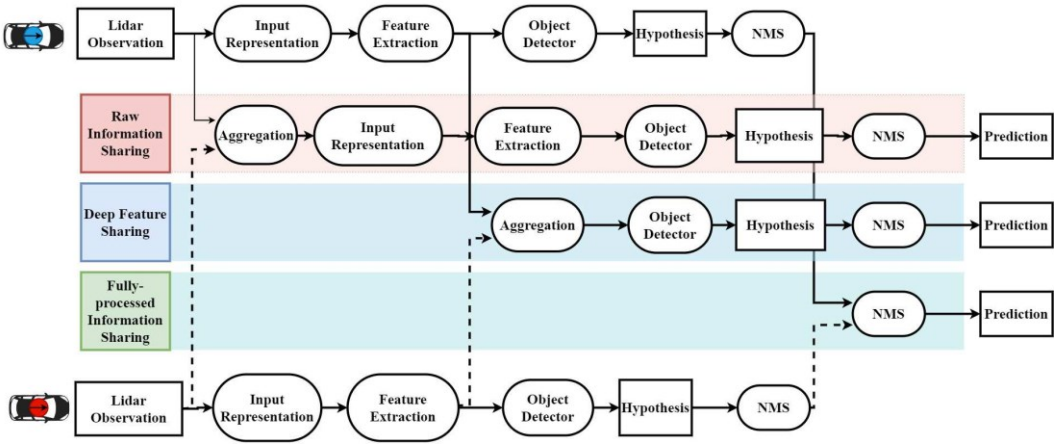


Figure 1.2 Overview of LiDAR based cooperative object detection methods [22]

1.4. Deep Features Information Sharing

In general, the proposed architectures for extracting deep features (and then fusion the shared information) mostly consist of the following modules:

1.4.1. Sensory to Input Representation Module

This module represents the information gathered from volumetric sensors (such as LiDAR). This representation determines the intrinsic trade-off between computational efficiency and fusion performance. Speaking of the performance, it is possible to project the sensor information onto 3D voxel tensors to prevent loss of information. However, the computation in this 3D space is based on using 3D convolutional neural networks, which is a very costly computation method. Since the height range of the object lying on the planar surface does not significantly affect the perception of the vehicle about its environment, it seems the projection of the 3D point cloud to a 2D map, while not significantly affecting the performance, could reduce the computation complexity. The standard 2D projection for fusing the sensor data is the bird eye view (BEV) which is an inverse perspective mapping that creates a top-down view of the captured environment by mapping 3D point-cloud pixels to a 2D map. This projection is a common approach that does not diminish notable performance in object detection. One of the important BEV projection features is not changing the size of the objects with respect to their distance from the observer. This advantage makes BEV an appropriate representation in cooperative methods where data comes from different viewpoints of different vehicles. In the following Figure 1.3 and Figure 1.4, this projection is illustrated.

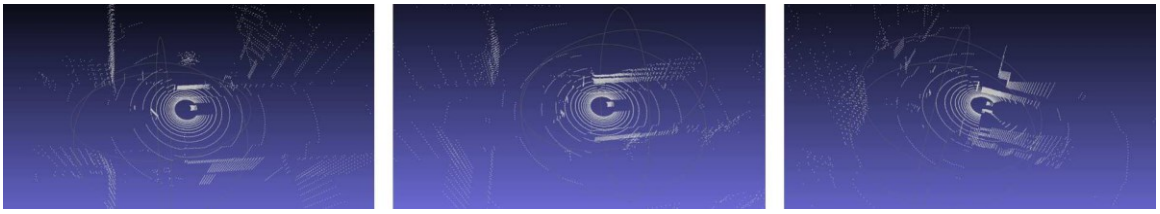


Figure 1.3 Point-cloud map generated by cooperative LiDAR Data Sharing [22]

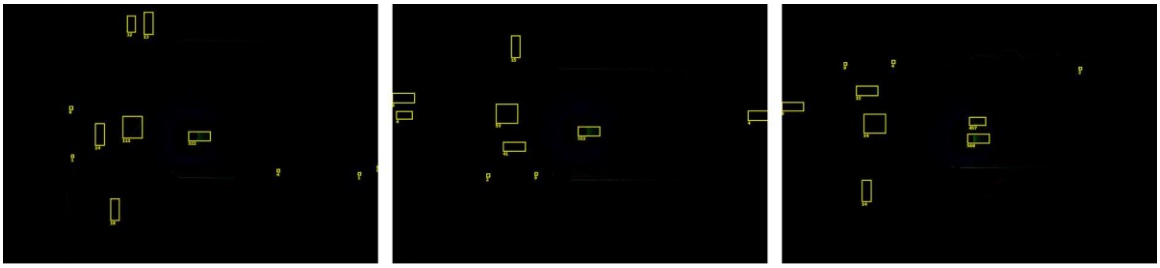


Figure 1.4 BEV representation of the Figure 1.3 point-cloud [22]

1.4.2. Feature Extraction Component

The feature extraction module is a neural network (the customary one is a convolutional neural network that takes BEV 2D image as the input) that has been trained to project the sensory input to the intermediary state space. The parameters of this neural network are trained based on the objective functions designed due to the desired fusion goal. So, the parameters are optimized with respect to the desired task. The number of sub-sampling layers in the neural network determines the compression rate. Consider that there is still a trade-off between the size of the feature map and performance. In the way that more sub-sampling layers lead to less feature map size and less efficient object detection performance. Therefore, the neural network structure affects the communication requirement of the cooperative system in both the number of channels and the down-sampling rate ways.

1.4.3. Transmissions and Receptions Systems

As we discussed, the neural network architecture determines cooperative systems' bandwidth requirement and performance. The size of the shared feature maps relies on the neural network design. The influential point about this entanglement is that the system could not be flexible enough to adapt to different bandwidth requirements. In other words, for inflexible architecture, on the transmitter side, lowering the feature maps' number of channels will only worsen the object detection performance without having any effect on bandwidth consumption. [27] is trying to make more flexible communication systems for lowering (or heightening) the required network capacity based on the size of the extracted feature maps.

In this paper, there are two considered banks of CNN encoders and decoders which (based on the bandwidth limitation) project the feature maps onto a lower dimension (with low volume) to adjust transmitting data with bandwidth limitations. From the

neural network point of view, the number of filters at the last layer of the encoder determines the size shared between vehicles. The following Figure 1.5 shows these two additional banks at the same previous structure:

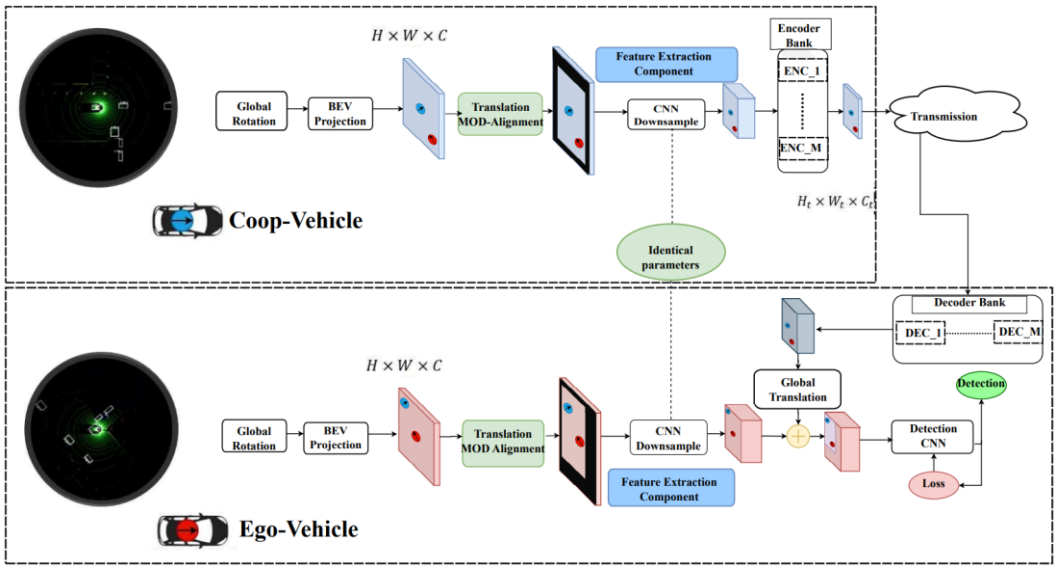


Figure 1.5 An overview of the bandwidth-adaptive feature sharing architecture [27]

Another flexible design is proposed in [31]. This paper addresses the available network bandwidth real-time limitations challenge by suggesting a point cloud feature based cooperative perception framework (F-Cooper). This framework supports voxel feature fusion and argues that it is possible to maintain detection performance and reduce the communication load transmission by selecting a subset of channels from the feature maps. This framework divides the original LiDAR detection area into a voxels grid. Apart from critical grids, the vast majority of voxels are empty since they are not holding any critical information, and consequently, sharing features map needs fewer resources.

1.4.4. Data Alignment

After receiving the extracted features map from different vehicles with different spatial distributions, it is necessary to have some translation and alignment operations to fuse the information gathered from different defined state spaces. While the alignment for wholly raw or full-processed data could be limited to simple translation and rotation transformations, the alignment process for DFS information requires more effort. In the clear-cut feature maps sharing, both sides have similar down-sampling layers in CNNs. Every pixel produced in the feature map represents a set of pixels in the input BEV image. In this scenario, the rotation alignment is taken place before transmitting by considering a global coordinate system:

$$X_w = X_e R_x(\alpha) R_y(\beta) R_z(\gamma) \qquad \text{Equation 1.1}$$

Where X_w and X_e are point representations in the global and local coordinate systems, respectively. R_x, R_y and R_z are also the rotation matrices on the desired axis. Afterward, since the BEV map is aligned before extracting the features and transmitting, the second phase of alignment is an image translation that is taken place on the received feature maps by the following Equation 1.2, :

$$\widehat{F_c}(x_f,\ y_f) = F_c(x_f + \Delta x,\ y_f + \Delta y) \quad \text{Equation 1.2.a}$$

$$\Delta x = \left\lfloor \frac{x_e}{s} \right\rfloor - \left\lfloor \frac{x_c}{s} \right\rfloor \quad \text{Equation 1.2.b}$$

$$\Delta y = \left\lfloor \frac{y_e}{s} \right\rfloor - \left\lfloor \frac{y_c}{s} \right\rfloor \quad \text{Equation 1.2.c}$$

That s demonstrates down-sampling rate. $(x_c,\ y_c), (x_e,\ y_e)$ are coop and ego vehicle locations in the global coordinate system, and $F(\cdot), \hat{F}(\cdot)$ are received and aligned received feature maps.

The alignment could differ for different down-sampling rates in coop and ego vehicles. This issue can cause some inconsistencies in the further aggregation stage. [27] addresses this challenge by introducing the translation MOD-alignment (TMA) method for mitigating the misalignment error caused by down-sampling. In this method, zero-padding is imposed on feature maps before down-sampling to avoid any potential localization error. At this agnostic (to the position of the coop and ego vehicle locations) method, the padding and down-sampling are formalized:

$$(p_l, p_t) \equiv (x_0, y_0)\ mod\ K \quad \text{Equation 1.3.a}$$

$$(p_r, p_b) \equiv (-x_1, -y_1)\ mod\ K \quad \text{Equation 1.3.b}$$

Where $[x_0, x_1] \times [y_0, y_1]$ determines the range of the input image in the global pixel-wise coordinate. K determines the down-sampling rate, and p parameters right, top, and bottom padding parameters. The following Figure 1.6 illustrates the Translation MOD-alignment procedure.

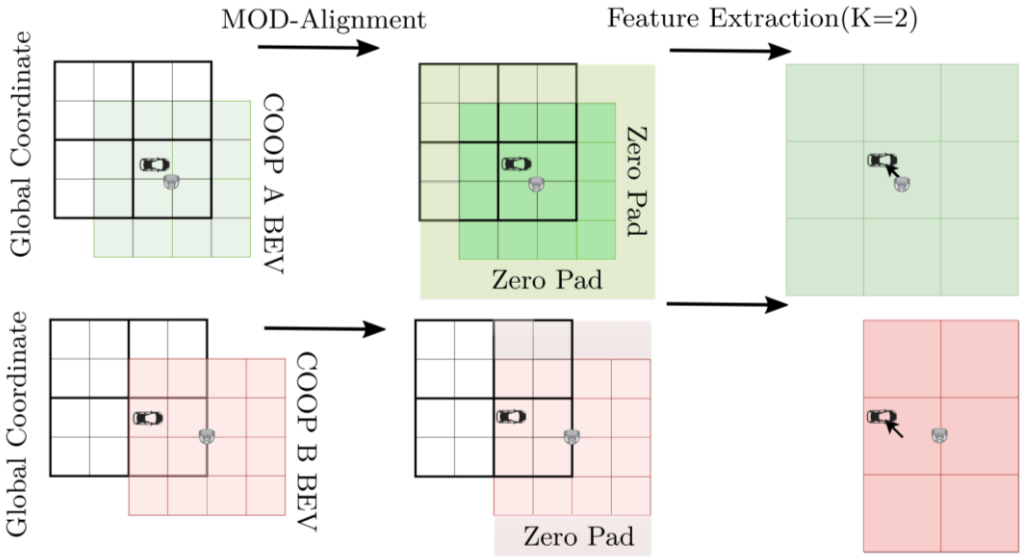


Figure 1.6 An illustration of translation MOD-alignment padding is applied on the input images [27]

1.4.5. Data Aggregation

Proposed aggregation methods are mainly based on the Fixels arithmetic operation. At [31], the features aggregation has been employed as an element-wise maxout scheme to fuse.

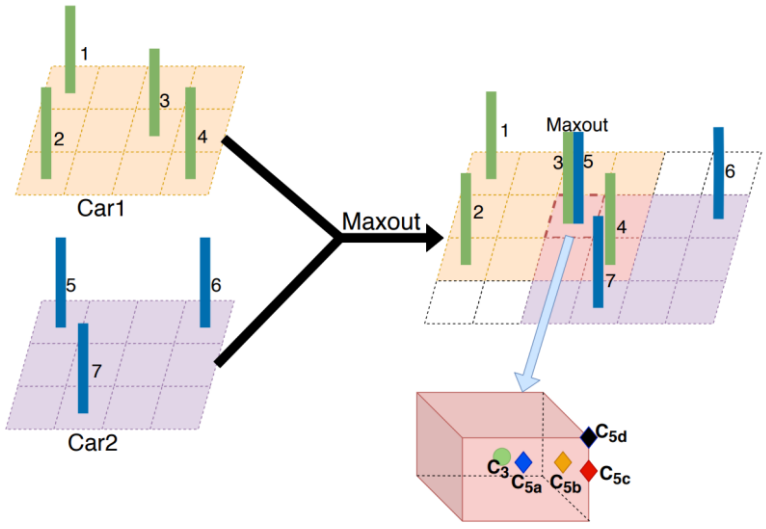


Figure 1.7 Voxel features fusion. Maxout function is used to fuse voxels. [31]

Another arithmetic operation that is used in [26] is a simple elementwise summation based on the idea that information acquired from the coop vehicle has the same level of importance as the ego vehicle's. Therefore, any arithmetic function should follow symmetric property concerning inputs (swapping the observation should not affect the fusion output.)

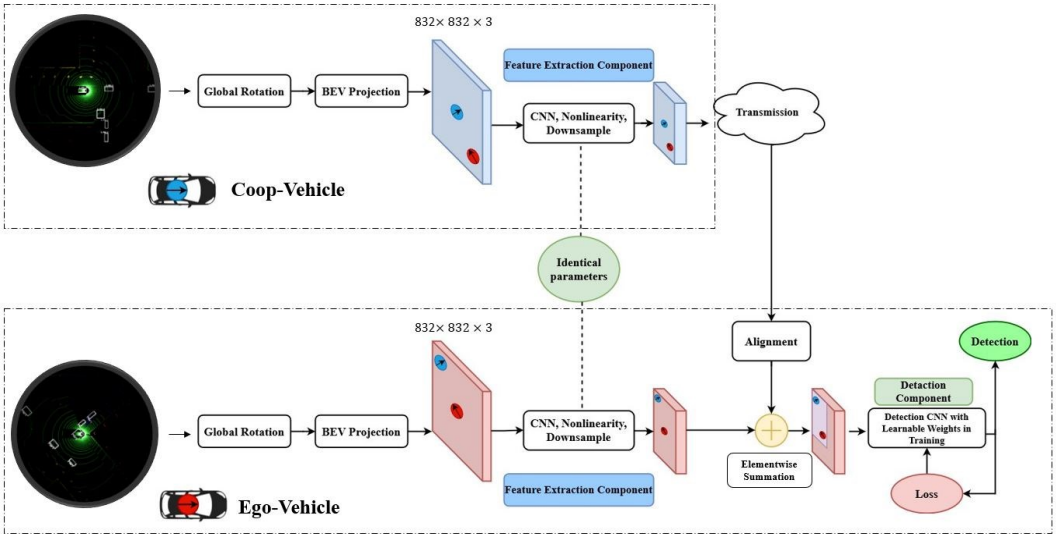


Figure 1.8 After aligning the feature maps, as demonstrated in ego vehicle box, the fusion is done by an elementwise summation. [26]

The other possible arithmetic operation which is used in [22] is the max-norm function for fusing. For the Fixel coordinates (x, y), and the i_{th} coop vehicle received (and aligned) feature-map, $V_i^{x,y}$ denotes the corresponding feature-map and the norm-max function is defined as:

$$\hat{V}^{x,y} = V_k^{x,y}$$
$$when\ k = \arg max_i \left(\left|V_i^{x,y}\right|\right) \qquad \text{Equation 1.4}$$

1.4.6. Training The Neural Networks

The training methods can seriously impact the performance of cooperative perception systems. As [22] discusses, there are two different ways to design and train two distinct neural networks for cooperation. The first approach was originally introduced in [23]. The fundamental theory behind this training is that implementing the same CNNs (with the same parameters) in both ego and coop vehicles, where fusion observations calculate the loss, could potentially improve the detection performance meaningfully. At [22], this joint training method is named Cooperative-Vehicle Training strategy (CVT). To describe this method, first, it is needed to review the possible existing neural networks in this deep feature sharing system, which are:

I. **Feature Extraction Networks:**
Mostly it is a CNN that gets the aligned BEV map as the input and converts it to the feature map of the vehicle's surrounding environment. Note that this network structure is identical in both coop and ego vehicles (transmitter and receiver) because both must project the input BEV maps onto the same feature space to make it feasible to have cooperative fusion. In [27] ,the used CNN structure is demonstrated in Table 1.1.

Table 1.1 The architecture of the proposed feature extraction network cited from [27]

Feature Extraction Network Parts
3x3x24 Convolution Batch-Norm Leaky ReLU(0.1)
Maxpool/2
3x3x48 Convolution Batch-Norm Leaky ReLU(0.1)
Maxpool/2
3x3x64 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x32 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x64 Convolution Batch-Norm Leaky ReLU(0.1)
Maxpool/2
3x3x128 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x64 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x128 Convolution Batch-Norm Leaky ReLU(0.1)
Maxpool/2
3x3x128 Convolution Batch-Norm Leaky ReLU(0.1)

II. Feature Encoding and Decoding Modules:

Although these modules are not entirely necessary in the system, as we said, it has been proposed by [27] to address the inconsistency in different bandwidth allocations. In designing the feature extractor network (that, as we mentioned, is identical for both coop and ego vehicles), the size of the feature map (and its channel numbers) depends directly on the specified network structure. In other words, the dimensions of the last layer of this convolutional network are designed concerning a constant bandwidth capacity that would not vary by time. Although this bandwidth maintenance makes it feasible to train these networks properly, it also leads to a rigidity of design in adapting to different bandwidth settings. A way-out solution for this rigidity is to train multiple features extracting networks with different channel sizes. This solution is not only inefficient in terms of memory usage, but the idea of reducing the number of channels to compress the features would worsen object detection performance. As we mentioned earlier, the proposed adaptive solution [27], with remarkable mitigation for performance diminishing, is creating a bank of encoder/decoder pairs that map actual feature maps onto a lower dimension (encoding), then driving back (decoding) the received encoded data at ego vehicle side to the original encoded feature-map

space. The number of filters at the last layer of the encoder network determines the size of transmitting data which means it could be possible to adapt the size of transmitting data with the allocated bandwidth resource by choosing the proper encoder. The last layer of the decoder has the same number of filters as the last layer of the feature extraction network, and the performance shrinking is significantly less than lowering the feature-map available channels.

III. Features Aggregation and Object Detection Network:

The final stage (after decoding, globally aligning, and aggregation) is to detect the targets in the environment by utilizing an object detection CNN module that gets the aggregated feature-map as the input. At [27] the suggested architecture is:

Table 1.2 The architecture of the proposed object detection network in [27]

Object Detection Networks Parts
1x1x128 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x256 Convolution Batch-Norm Leaky ReLU(0.1)
1x1x512 Convolution Batch-Norm Leaky ReLU(0.1)
1x1x1024 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x2048 Convolution Batch-Norm Leaky ReLU(0.1)
1x1x1024 Convolution Batch-Norm Leaky ReLU(0.1)
1x1x2048 Convolution Batch-Norm Leaky ReLU(0.1)
3x3x1024 Convolution Batch-Norm Leaky ReLU(0.1)
1x1x20 Convolution
Output 52x52x20

Since the explained framework consists of various encoder/decoder pairs, the parameters of Feature extraction and object detection components must be trained to be compatible with all available encoder/decoder pairs. These modules' compatibility needs to train all the modules simultaneously (reminding again that the ego and coop vehicles have the same components with the same parameters for feature extracting). During the training process in [27] to have compatible training for different encoding/decoding pairs, they randomly select an encoder/decoder function from the available bank. The gradients in the back-propagation step (for updating the network elements) are calculated with respect to cooperative vehicle observations and the feature extractor and encoder-decoder component parameters.

Function g (feature aggregation function) could be defined as:

$$g\big(f(Z_1;\theta), f(Z_2;\theta)\big) = f(Z_1;\theta) + h(f(Z_2;\theta);\eta) \qquad \text{Equation 1.5}$$

Where Z_1, Z_2 respectively are ego and coop vehicles observations. θ is the feature extractor network parameters. η is the encoder/decoder pairs parameters.

Accordingly, the partial derivative with respect to parameters θ (feature extractor network parameters), η (encoder/decoder pairs parameters) is calculated by:

$$\frac{\delta g\big(f(Z_1;\theta), f(Z_2;\theta)\big)}{\delta\theta} = \frac{\delta f(Z_1;\theta)}{\delta\theta} + \frac{\delta h(f(Z_2;\theta);\eta)}{\delta\theta} \times \frac{\delta f(Z_2;\theta)}{\delta\theta} \qquad \text{Equation 1.6.a}$$

$$\frac{\delta g\big(f(Z_1;\theta), f(Z_2;\theta)\big)}{\delta\eta} = + \frac{\delta h(f(Z_2;\theta);\eta)}{\delta\eta} \qquad \text{Equation 1.6.b}$$

For the lost function, it is possible to use the proposed lost function in [32] that is defined for predicting multiple bounding boxes:

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] \\
&+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
&+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{ classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \qquad \text{Equation 1.7}$$

One more considered neural network and training method comes from [31]. It introduces end-to-end 3D object detection leveraging feature-level fusion. The proposed framework supports two fusion methods: voxel feature and spatial feature fusion. As it is illustrated in the Figure 1.9, the first step in both methods is voxel features extraction for converting the LiDAR input data to a voxel features table. For generating the voxel features, this paper uses the VFE layer of VoxelNet [33]. The next neural network level defines these methods' main differences. In the first method, the sets of voxel features (of the coop and ego vehicles) are fused first, then after collecting all features in the ego vehicle, the spatial feature maps would be generated after voxel tables aggregation by coming networks. Therefore, it is named Voxel Feature Fusion (VFF). On the other hand, for the second method, after extracting the voxel features, the initial spatial features are extracted locally and individually in vehicles. Then, these

local extracted spatial features would be aggregated in the ego vehicle to generate the final feature map.

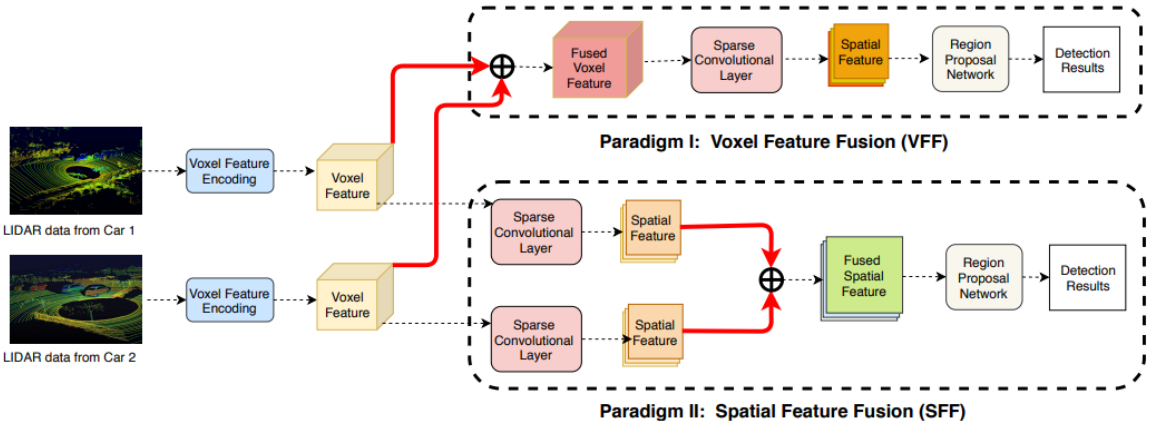


Figure 1.9 Architecture of the feature based cooperative perception [31]

The other similar block that is used in both paradigms is the Region Propose Network (RPN), which has been placed in the last stage. After passing the spatial features to this network, the generated outputs would be the locations and probability scores of the proposed regions. The architecture of applied RPN is obtained from [33], which proposes this network as a top-performing object detection framework. In that work, the RPN is combined with the feature learning network and middle CNNs in a trainable pipeline for applying an E2E training procedure. This network has three components of fully convolutional layers. The first one down-samples the feature maps (Input) by half, followed by a sequence of convolutions of stride. After each convolution layer, the operations batch normalization (BN) and rectified linear unit (ReLU) are applied to the data. After the down-sampling and projecting to lower dimensions, the network up-samples the output of every block to a fixed size and then concatenates them to construct the high-resolution feature map. The output defined in [33] is like the output mentioned in the [31], which is regression and probability scores map. The architecture of this proposed network could be illustrated as follows (Figure 1.10):

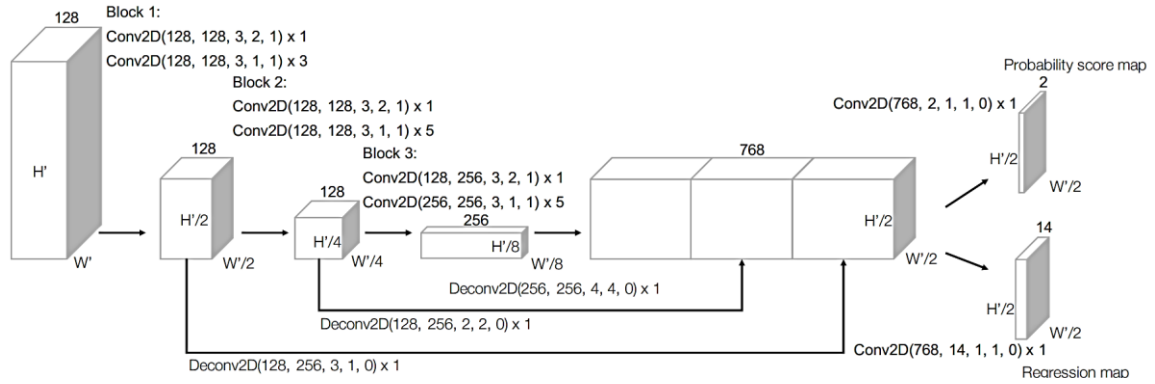


Figure 1.10 Region Proposal Network (RPN) architecture proposed in [33]

The significant difference between the F-Cooper structure with the past network architecture is the training procedure imposed on the system. The initial state in F-Cooper is based on the pre-trained single vehicle object detector CNNs used in this system. The fused spatial features (that could be aggregated by both SFF and VFF methods) are passed to an RPN network, which is a vehicle object detector and has been pre-trained beforehand. Since the training process for the RPN network is being done individually in each vehicle, in [22], F-Cooper is called the single-vehicle training strategy (SVT). Remember that the cooperative-vehicle training (CVT) strategy training method is based on the identical Feature Extraction Networks for all alliance vehicles.

1.4.7. Obtained Results

At this point, we are reviewing the practical results of [22] for all combinations and methods to evaluate the scalability and noise sensitivity of all various approaches. The required data is created by simulating the desired observations. This tool generates simultaneous observations from the same scene using an open urban driving simulator, CARLA [34]. This simulation advantage is the capability to gather synchronous measurements such as RGB images, LiDAR point-clouds data, and ground-truth information (such as GPS information and bounding boxes). The main goal of this perception is to detect pedestrians and vehicles in an urban area (with all possible existing objects). The connection graph among LiDAR-equipped vehicles in this environment is random and must fulfill some conditions. More specifically, in the paper [22], LiDAR observation is randomly paired with another observation on the condition that the observers have at least a mutual target, which is placed under forty-meter. This evaluation analyzes all three primary information sharing methods, which are Raw Information Sharing (RIS), Deep features sharing (DFS), and Fully processed Information Sharing (FIS). In the RIS method, the raw data is aggregated from all coop vehicles and ego observations and then would be fed into some CNN architecture. For DFS, at each coop vehicle, the raw observations (BEV maps) are firstly fed into the local Feature Extraction Component (FEC) networks to extract the deep features at different vehicles. These features are aligned, transmitted to the ego vehicle for aggregation, and then passed to the final Object Detection Module (ODM). In this paper, the implementation of the FIS approach is based on first transferring the final processed data (extracted and processed independently by coop vehicles) to a defined hypothesis space, then sharing the final interpretable hypotheses. This method is named Hypothesis Sharing Method (HSM), where the generic form for the hypothesis vector H is formulized:

$$\widehat{H} = \left[\delta\widehat{x}, \delta\widehat{h}, \widehat{\omega}, , \widehat{\rho_c}\right] \qquad \text{Equation 1.8}$$

Where $\delta x, \delta h, \omega, \rho_c$ notations represent localization, shape, orientation, and predicted classification, respectively.

In this experience, the specified indicator for assessing performance is average precision (AP), which is calculated based on the Intersection-over-Union (IoU) threshold of 75%. Furthermore, this paper also tried to analyze the detection sensitivity with respect to GPS noise levels. For each sample, a positioning error, with uniform distribution for direction and white Gaussian distribution magnitude, is added to the ground-truth position of cooperative vehicles. The error magnitude at this evaluation varies from 0 to 2.4 meters. The other more critical evaluation in this paper is the Scalability test. This examination's primary purpose is to investigate the efficiency of vehicle cooperation with respect to the number of LiDAR-equipped vehicles in the simulated urban area. It is possible to describe the main goal as the performance inclining (or declining) with regard to participating entities' numbers and adjustments.

In the first study, After training the network based on SVT strategy (and applying trained parameters for all different approaches), In order to evaluate different DFS cooperative methods along with RIS and FIS methods, The average precision (AP) was computed for detecting vehicles and pedestrians in different scales and GPS noises. The results for the study's first phase, GPS noise sensitivity, which is summarized in the Figure 1.11, indicate that DFS methods have fairly more firm performances concerning localization noises that appear because of GPS noise. For pedestrians' detection, this robustness shows itself even better.

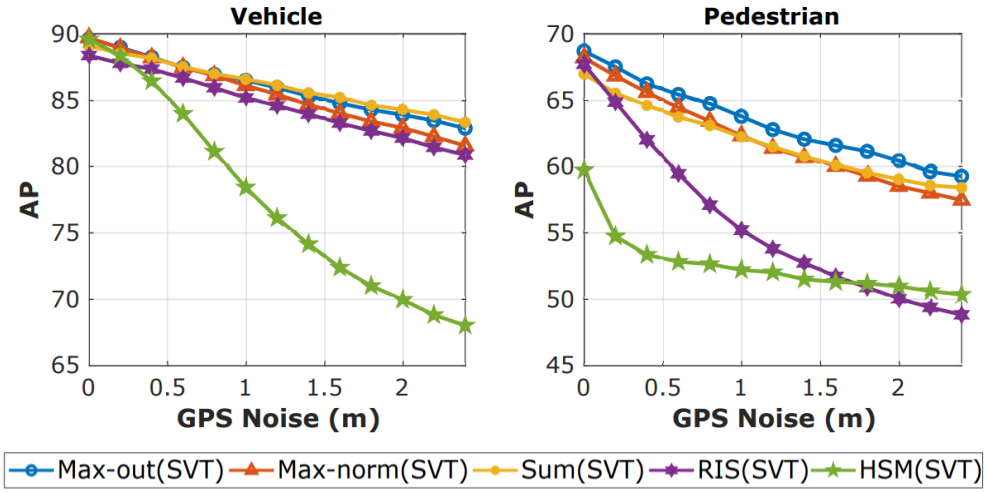


Figure 1.11 GPS noise sensitivity test and the effect of different information aggregation functions [22]

The next phase, which is about the number of coop vehicles and cooperation scalability, illustrates that the DFS method gains faster performance as the number of cooperative vehicles increases when it uses max-norm as the arithmetic operations for aggregation. Furthermore, this result shows a strict disruption in DFS performance for the element-wise summation aggregation method when the number of coop vehicles increases. However, this decline is not so unexpected because the neural networks are not trained based on element-wise aggregation. Subsequently, the feature maps and

the corresponding CNN layers are not adjusted for this fusion method. These results also are figured below:

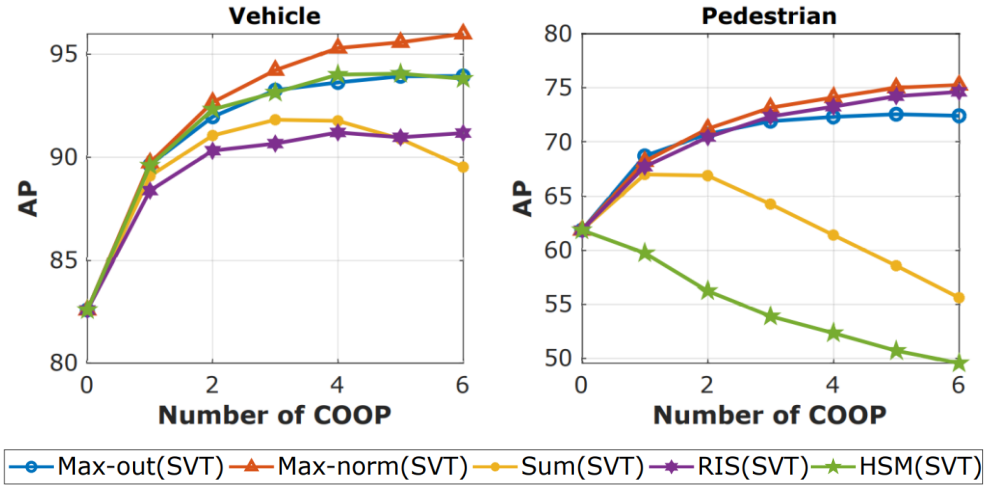


Figure 1.12 Scalability test and the effect of different information aggregation functions [22]

Due to the performance disruption that occurred for element-wise summation in the previous test, the RIS method could be a suspicious and unsure case for scenarios with a large number of cooperations. Because in the RIS method, the aggregation of raw data acquired from cooperative vehicles is equivalent to the summation of received BEV maps. This uncertainty is the primary object of the future study, demonstrating the effect of two different training strategies (SVT and CVT) for RIS and the DFS information sharing methods when the aggregation method is the element-wise summation. This study confirms this by claiming that Cooperative-Vehicle Training (CVT) is not only able to resolve this disruption, but also there is a performance enhancement for larger cooperation scales. These test results are illustrated below Figure 1.13:

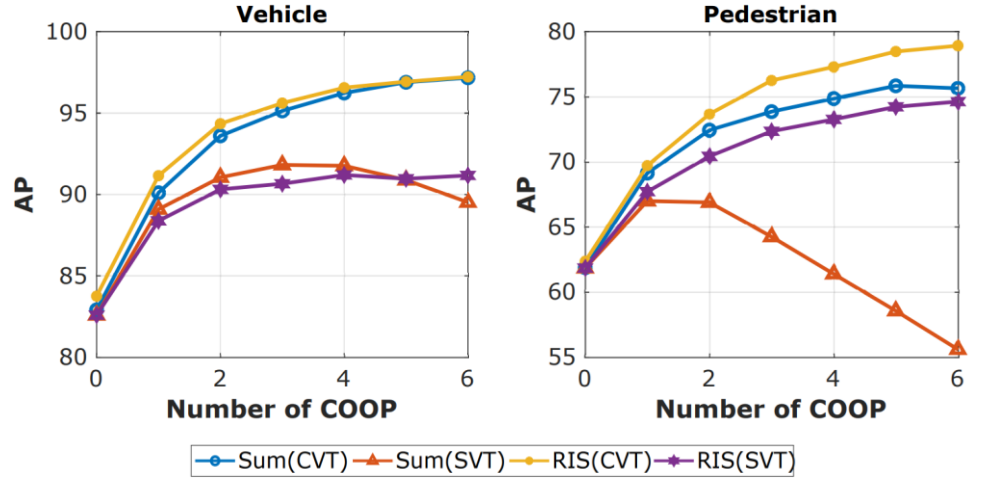


Figure 1.13 scalability test for RIS (CVT & SVT) and DFS (CVT & SVT) methods for summation aggregation methods [22]

In order to present a comprehensive perspective of the best performing network structures, [22] provide the following study to give a conclusive insight into a variety of decent cooperative perception approaches. Regarding the scalability test, it looks like RIS has the best performance. Later, the second performance belongs to the DFS method with an element-wise aggregation function that performs moderately worse than RIS. While the DFS + max-norm aggregation function also shows promising results while not requiring cooperative training. The comparison of the best scalability attained results is shown in Figure 1.14:

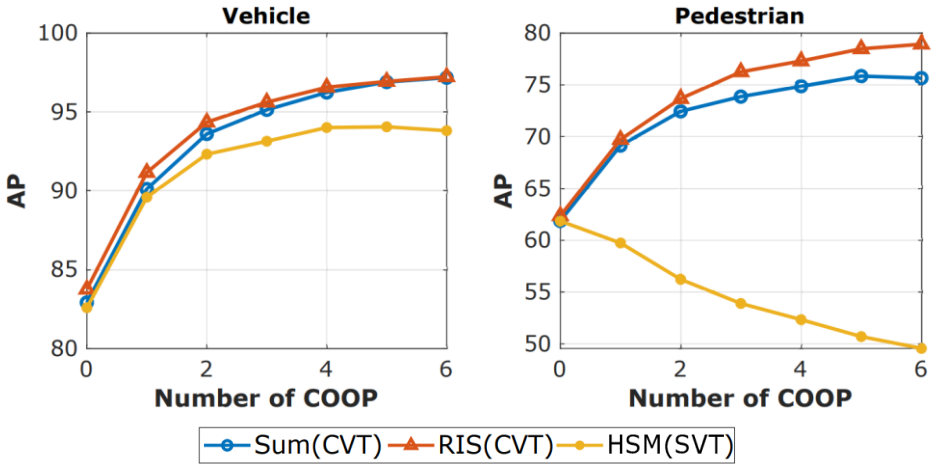


Figure 1.14 The comparison of best performing choices of scalability test [22]

Another valuable result that is achieved from the GPS noise sensitivity test is DFS method robustness, especially in pedestrian detection performance. This robustness is visible and noticeable in Figure 1.15:

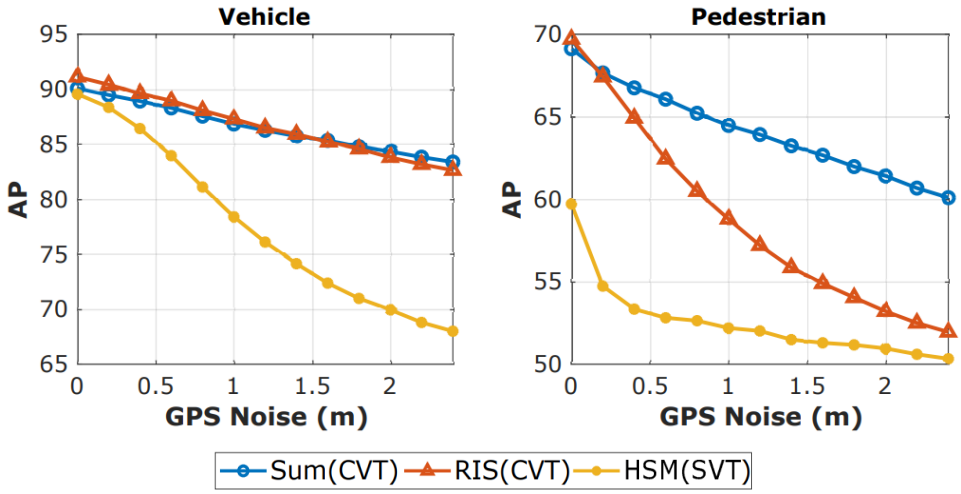


Figure 1.15 The comparison of best-performing choices of GPS Noise Sensitivity Test [22]

A vital issue in this comparison that has been evaluated in [22] is the poor performance of the HSM (FIS) method. This insufficiency shows itself in both GPS noise sensitivity and scalability tests. There is even a drop in HSM detection performance by increasing the number of cooperative vehicles. This drop is even more significant in pedestrian

detection. In order to analyze this performance dropping, the paper also makes the previous comparison concerning two other indicators; These two additional indicators are precision and recall. Even for precision indicators (shown in the Figure 1.16 and Figure 1.17), it is still possible to detect the significant performance falling as GPS noise magnitude or the number of cooperation increases. The paper identifies the reason for this drop effect because the Non-Maximum Suppression technique is used to merge the shared hypotheses in HSM single-shot networks. These sensors have a bounded number of hypotheses per cell. In the RIS and the DFS approaches, since detecting hypotheses happens after data aggregation, the total number of hypotheses (which would be used for calculating the AP indicator) is not dependent on the number of coop vehicles. While for the FIS method, the cooperative vehicles share the derived hypotheses with the ego vehicle, which means that the number of hypotheses depends on the number of vehicles.

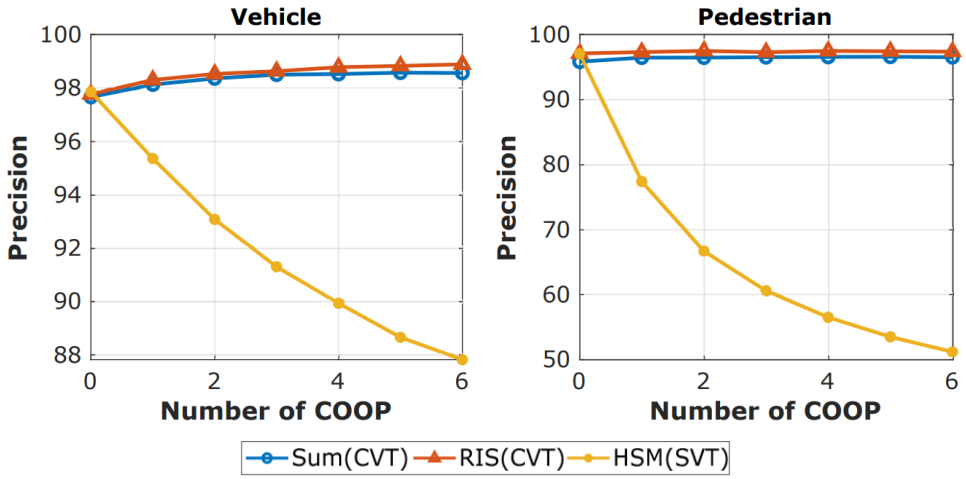


Figure 1.16 The comparison of best-performing choices of scalability test with respect to precision indicator [22]

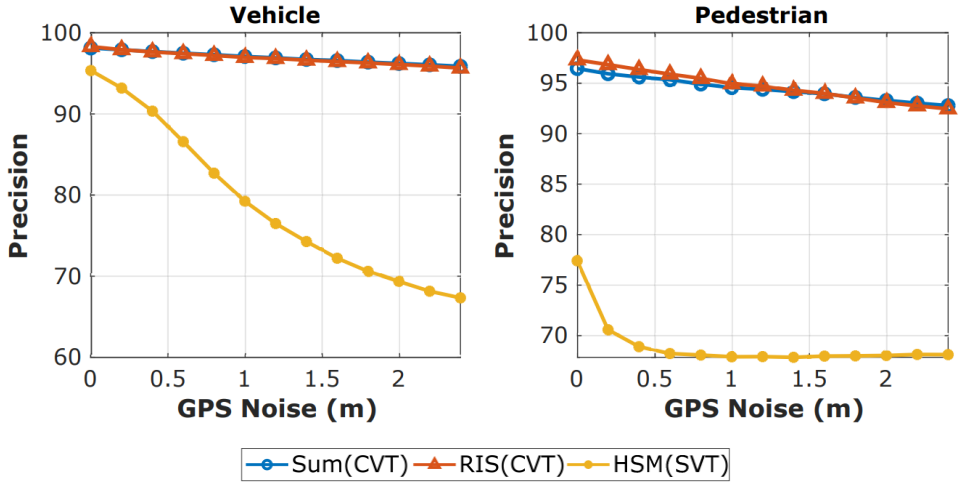


Figure 1.17 The comparison of best-performing choices of GPS noise sensitivity test with respect to precision indicator [22]

On account of the above explanations, it seems that there must be an association between several participants and detection inadequacy. The paper gives two possible factors that have the potential to excuse this relation:

1. As more cooperative participants share their hypotheses, if there is a non-acceptable and poor merging function, the number of false-positive cases might also increase. This means a more false-positive ratio. This factor also can explain the reason for the more dramatic precision drop in pedestrians' detection.
2. The second factor also is related to the non-maximum suppression merging technique and the fact that if coop vehicles share false-positive hypotheses, NMS will merge such false hypotheses. Consequently, increasing the number of cooperative vehicles could increase the existing false-positive hypotheses in the aggregated shot, leading to the accumulation of false-positive hypotheses in final detection. The reason for not observing this issue in RIS and DFS is that besides the number of participants, there always is a limited number of proposed hypotheses per cell. Moreover, these fusion methods would lead to false positive instances purging.

After indicating these two factors, the paper suggests that by developing the non-maximum suppression (NMS) function, it would be possible to get better precisions for all sharing methods because all applied methods utilize an identical NMS function. Overall, it must be indicated that the FIS method's ineffectiveness in most cases should first be referred to NMS method's potential flaws. It is vital to redesign the NMS function to address the detected failings adequately.

Nevertheless, it is pretty evident that projecting the raw sensory data to an intermediate space of deep features and then sharing it among all participants could be a promising approach for sharing the multimodal sensing data with respect to allocated bandwidth and accepted AP shrinking. More importantly, although the RIS method achieves the best AP, DFS is a more robust sharing method as localization noise increases. As pointed out, it is even less sensitive to GPS noise than the RIS method.

Although [22] eventually claims that using DFS method could lead to less bandwidth demands, it does not present any data or numbers to verify this claim. Not thinking about the amount of required transmitted data in different RIS, DFS, and FIS sharing methods could be one of the main deficiencies of this article which prevent us from reaching a certain reasoning to switch from sharing raw data to sharing feature maps which have been extracted with the assistance of deep neural networks.

In order to get a sense of the size of sharing data in different methods and possible shrink bandwidth usage that could be earned by switching to semantic feature sharing, it is worth noting and inspecting the recent papers. Since we mentioned [31] research earlier, to talk over its utilized networks briefly, it seems reasonable to come back to this paper again and mention its achievements in terms of earned accuracy and

transmitted data size. As we mentioned, this paper proposed two different methods for sharing spatial features, Voxel Feature Fusion (VFF) and Spatial Feature Fusion (SFF). For analyzing detection precision, the paper tries to evaluate these two methods in two different scenarios: "Near" and "Far." the "Near" and "Far" cut-off is 20 meters from the observer vehicle. Based on using the Intersection over Union (IoU) threshold at 0.7, this paper presents the precision accuracies of VFF, SFF, raw data fusion, and no fusion forms, expressed in Table 1.3. For no cooperation, we observe that although Car 1 achieves a good "Near" detection precision, it falls off sharply in precision in their "Far" detection. About deep features sharing method testing, in road scenarios, the VFF method can achieve a precision similar to raw data fusions. This likeness is interpreted as the VFF capability for near object detection. This precision is so close to the raw data fusion method as well.

Table 1.3 Precision comparison for different methods with respect to average precision (AP, in %) [31]

Scenario	Dataset	Baseline without fusion		VFF		SFF		Raw data fusion	
		Near	Far	Near	Far	Near	Far	Near	Far
Multi-lane roads	KITTI	63.22	22.37	77.46	58.27	50	57.14	77.46	71.42
Road Intersections	T&J	78.37	19.6	80.21	72.37	73.68	53.33	80.21	72.37
Parking Lot1	T&J	58.33	33.33	66.67	62.54	66.67	33.33	66.67	70.58
Parking Lot2	T&J	66.67	18.85	72.25	46.42	72.25	25	75.5	50
Parking Lot3	T&J	N/A	45.81	N/A	66.41	N/A	66.41	N/A	66.41
Parking Lot4	T&J	100	N/A	100	48.83	100	33.33	100	48.83

In comparison between the SFF and VFF methods, there is a severe difference between the "Near" and "Far" precision of the SFF method. While SFF cannot overtake VFF in terms of precision, it can still outperform the baselines (no fusion) in most scenarios. This moderate performance is explainable due to the spatial features sparser table than both voxel features and raw data. So, it is sensible to have better precision in the denser regions, which includes near vehicles situations.

To highlight the differences in VFF and SFFs performance for near and far scenarios, the paper demonstrates their performance in the form of cumulative distribution functions of increased detection precision. Based on this demonstration (shown below) and for the far category, VFF can potentially increase its detection precision by 40% for

about 85% of the time. In contrast, for the SFF method, the detection precision increase is not as impressive as VFF performance, and there is a notable gap between their detection accuracy improvement for constant CDF.

Regarding the near category, as we can see, SFF can increase detection accuracy as much as VFF increases (if not better in some cases). For example, both have about a 35% chance of increasing detection precision by 30%.

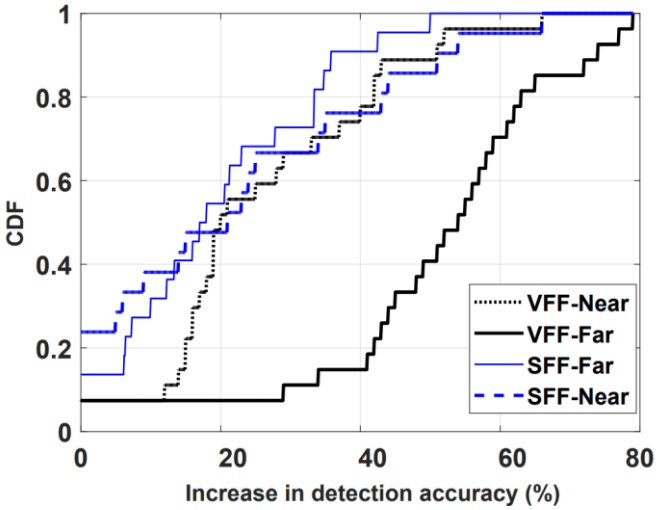


Figure 1.18 Cumulative distribution function vs. detection precision improvement for VFF and SFF methods in two main different far and near scenarios [31]

Another significant point in Figure 1.18 is the VFF meaningful performance for a far category than near one. This outperforming could be explained by extra Voxel features collected in far-type observation. Through fusion, the normal skipped points in the distance are given by coop vehicle observation, enhancing the ego vehicle's detection result. As a trivial rule, this detection precision may increase even further with more participating vehicles.

1.4.7.1. Vehicles On-edge Computation and Transmissions [31]

If we go back to the main reason for reviewing the proposed F-Cooper Fusion Pipeline [31], it was investigating the deep feature extraction module used in this pipeline. As we saw in the previous section, based on defined indicators, sharing the extracted features with the VFF method and fusion volumetric observed data has almost the same results as sharing the raw observations of coop vehicles. Regarding this achievement, the other aspects needed to study are the required imposed computation at the edge points of coop vehicles to extract deep features and the feasible bandwidth reduction that could be gained by switching from the RIS to the DFS.

Speaking of bandwidth allocation, it is worth noting that there is some primary difference between sharing the extracted spatial features (SFF) and sharing extracted Voxel features in terms of data structure. In SFF, firstly, the LiDAR bird-eye view is passed to VoxelNet [33] to encode and compress it to the voxel features table, then

these features are passed to the local feature learning network to generate spatial feature maps. As is shown in Figure 1.19, spatial feature maps have C different channels where the channels indicate the corresponding kernel numbers used in the CNN feature learning network.

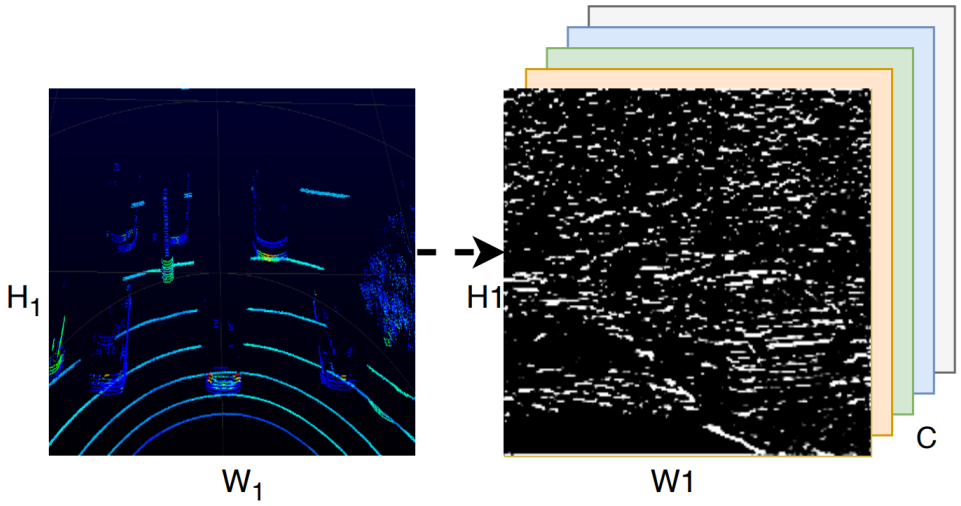


Figure 1.19 An illustrated example of projecting a LiDAR bird-eye map to spatial feature maps. H1 and W1 represent the size of the view, and C indicates the channels number. [31]

At F-Cooper architecture, the number of channels is set to 128. Since the fusion operation at the ego vehicle is being done channel-wise, it would be possible to only transmit a limited number of channel maps to the ego vehicle to minimize transmitting data by conveying only the most critical data.

To evaluate the performance diminishing with respect to the number of shared channels, the [31] has considered a constant case (which has been figured below, Figure 1.20) and analyzed Detection precision variations with respect to different numbers of selective channels on the SFF method.

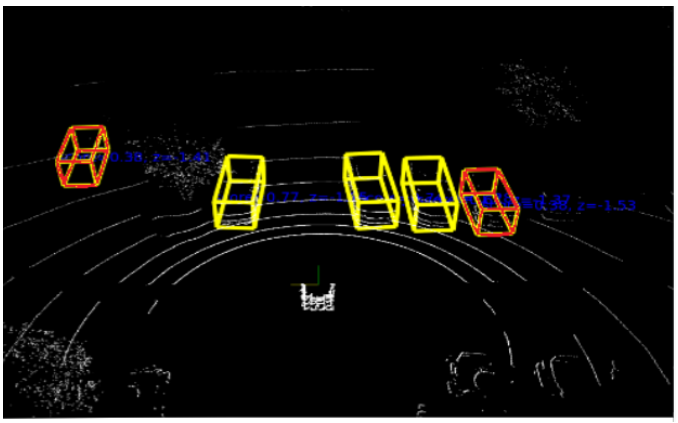


Figure 1.20 An experimental scenario in [31] which includes 5+1 vehicles

Based on this evaluation and the gained results, which can be found in the following diagram, 128 (0 to 127) channels represent the whole feature map. Channels 55-99 represent a range of vital channels providing the most useful data for SFF fusion.

Moreover, channels 95-99 represent a shrunk set of the most required and necessary channels to obtain some acceptable outcomes.

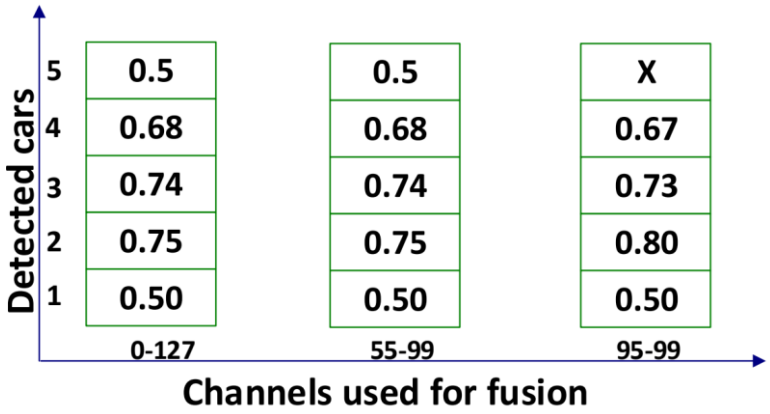


Figure 1.21 Detection precision for selective channels on SFF method [31]

This finding (which indicates that it is possible to decrease the number of transmitting channels by 96% with negligible detection precision disruption) is crucial for deploying some fusion strategies with respect to the available bandwidth and computation time.

The paper makes two comprehensive comparisons to assess each possible strategy's required volume and processing time. The paper explains that raw point cloud data is about 2 MB for assessing the required volume. This data volume could rise to 72 MB by converting to spatial features or reduce to about 1~1.3 MB by projecting to voxel features. However, due to the sparsity property of these maps, it is possible to compress these maps to less than 1 MB. In the SFF method, by compressing 55-99 channels, it is possible to get the highest compression (average 250 KB) results and the lightest strategy. There is no need to highlight this point in the SFF method is also possible to take the even more vital data by only selecting 95-99 channels information and reduce the volume dramatically. At the same time, SFF would still stay capable of achieving a similar precision. The Figure 1.22 displays this comparison:

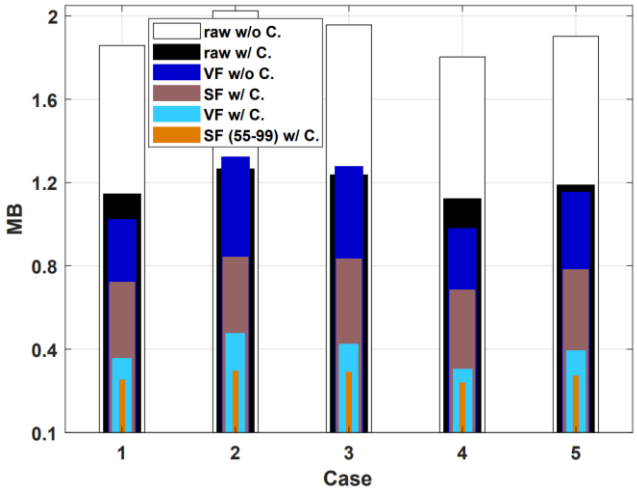


Figure 1.22 different fusion approaches and their corresponding average data volume

The Other comparison done in [31] is about the mandated times for different approaches. How different strategies perform regarding time consumption. This comparison matters because the ego vehicle may need to process and treat multiple requests simultaneously, and this processing time in both the ego and coop vehicles may make bottleneck by agents increasing. The total time includes processing, transmission, and object detection needed terms. Considering the uncompressed raw data transmission as the baseline, the total time for the SFF fusion strategy is close to the baseline, which both are about 0.8~1 second. Both SFF and VFF methods with compression could lower the total time to half with respect to the baseline. At the same time, both only need less than 0.1 seconds to send (and receive) the processed data. Probably, without considering the receiving and fusion processes in the destination part, the proposed methods are feasible to be employed to get reliable enhancing perception techniques.

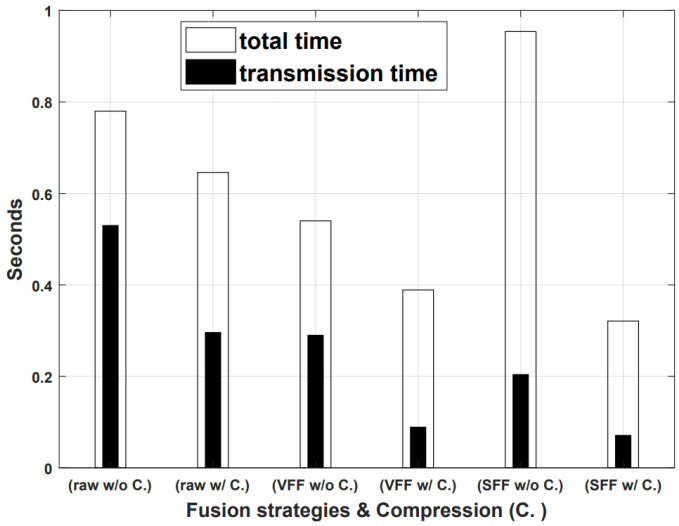


Figure 1.23 Time consuming comparison on different fusion strategies [31]

Another more recent paper that addresses fusion information sharing is [35]. This paper analyses a new pipeline architecture for aggregating Deep features information from multiple connected vehicles for fusion purposes. In their investigations of this new pipeline, they also discussed deep feature extraction compression rates and discussed it in detail. In the following, this pipeline is inspected quickly, and its data volume performance will be presented. As same as all reviewed networks, the OPV2V network also consists of these major components: Metadata Sharing and Feature Extraction, Compression and Feature sharing, Attentive Fusion, and Prediction Header.

I. Compression and Feature Sharing:

For compressing the raw LiDAR data and then transmitting it, this paper uses a pair of Encoder/Decoder modules to compress the shared information. The primary structure of the encoder architecture is based on the [36] proposed convolution/deconvolution network. The convolution network consists of a series of 2D convolutions and max pooling networks that extract features and transform them into a multidimensional feature table. The composing feature table is the encoded data sent from the coop to the ego vehicle as extracted deep features. On top of the ego vehicle receiving network, there is a multi-layer deconvolution network to recover the compressed information. The recovered map would then be passed to the Attentive Fusion module fuse decompressed received features. The utilized encoder/decoder networks should have an architecture so similar to the following network that has been proposed in [36]

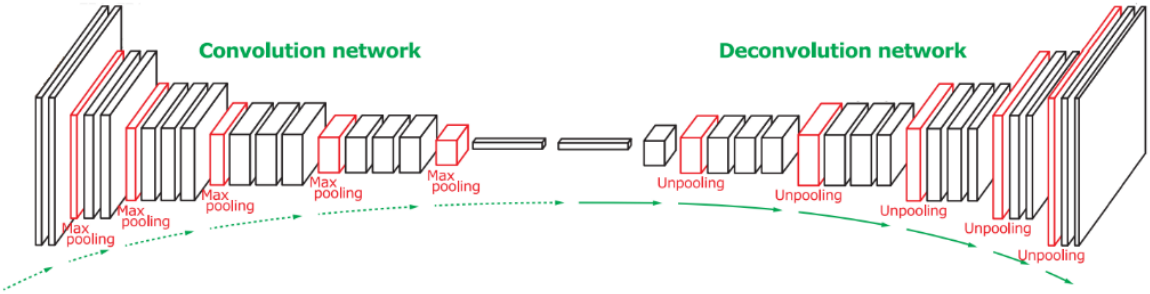


Figure 1.24 The Possible utilized architecture for Encoding/Decoding. Inspired from [36]

II. Attentive Fusion:

This module is reached from the self-attentions model, composed of fusing the decoded features and the ego vehicle observations. The interesting thing about this fusion module is that unlike the previous study, which used the element-wise summation as the aggregation method, this study tries to create a graph-based method to reason the interactions for better feature aggregations. This paper claims that since each of the received feature maps to the ego vehicle corresponds to certain spatial areas in the original point clouds, the simple (weighted or unweighted) sum of features will break spatial correlations. So, creating a local graph for received feature vectors (that edges are built from connected vehicles in the same spatial locations) and implementing a self-attention model could result in a good fusion network considering the gathered data correlation.

III. Prediction Header:

The features fused in the Attentive network would be passed to the prediction header to detect obstacles and vehicles, generate bounding boxes, and determine associated probability scores. Figure 1.25 shows all these main components in a comprehensive form as the OPV2V proposed architecture.

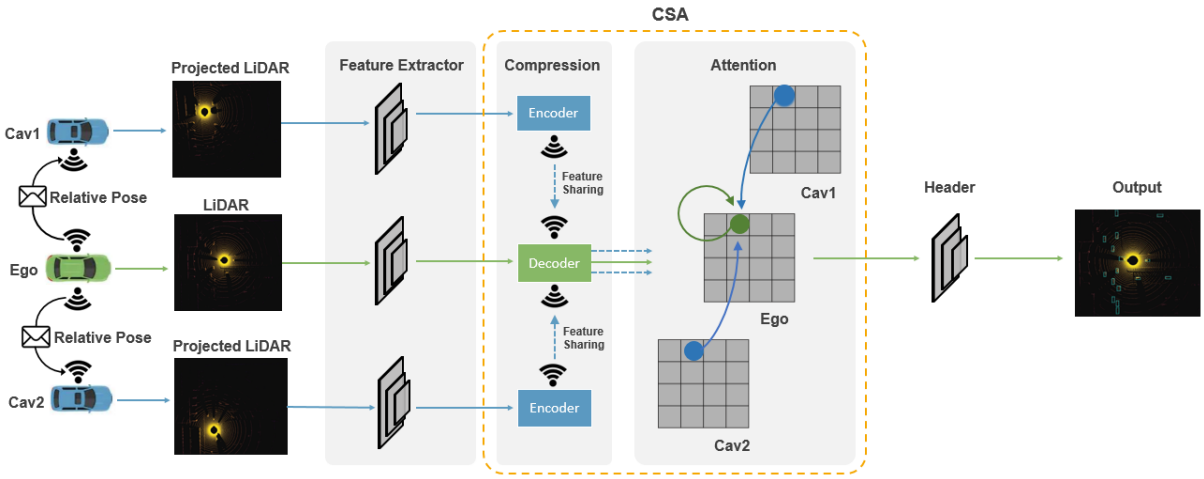


Figure 1.25 The Proposed OPV2V Architecture [35]

After this brief explanation of implemented networks, it is time to see the pipeline experiments. Ultimately, the two more important comprehensive comparisons could be made for evaluating the AP variation with respect to the number of participants as well as evaluating the AP variation with respect to the different data compression rates for different information sharing methods.

1 --Average Precision With Respect to the Number of Coop Vehicles:

For a complex defined area with 150 scattered vehicles, some of them would be selected as cooperative vehicles that start sharing their observations with ego vehicles. For different combinations and a different number of coop vehicles, this simulation has been repeated, For the average precisions (AP) indicator at the Intersection-over-Union (IoU) threshold of 0.7, the relation between the number of Coop vehicles and indicator has been figured, at Figure 1.26 the AP has a positive correlation with the number of coop cars. For a sensible reason, it seems that by increasing the number of coop vehicles, we will experience an indicator enhancement with a decreasing rate of improvement. In other words, the AP increasing rate for coop vehicles more than five becomes lower for coop vehicles. This sensible reason could be that after a certain amount of coop vehicles, the union of shared perception almost covers most of the ego vehicle's blind spots, and extra coop vehicles would not have such useful new information that leads to notable AP improvement.

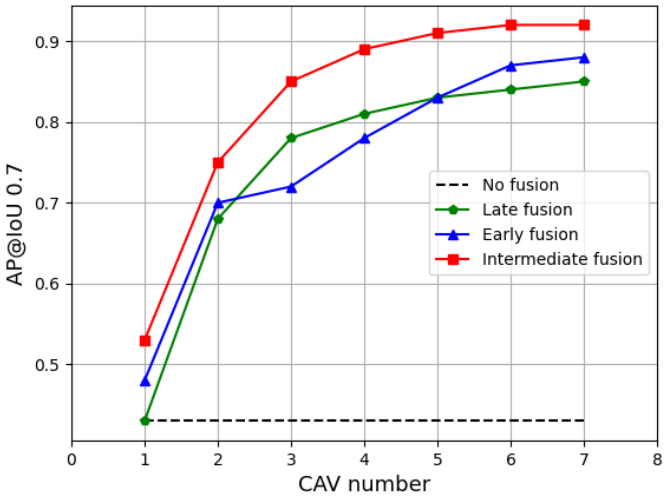


Figure 1.26 The AP at IoU=0.7 with respect to the number of coop vehicles [35]

2 -- Average Precision With Respect to the Size of Coop Vehicles Transmitted Data:

As we mentioned earlier, due to the structure of the utilized neural network, it is possible to encode and project the input raw sensory data to different feature maps with different compression rates. By modifying the number of layers in encoder/decoder modules, OPV2V can reach different distinct compression rates. The paper does not mention the details of these network modifications and the training process for each compression rate. However, they seem entirely separated from each other, and all need their own separated memories and networks. As we reviewed in [27] one of the suggested approaches to compose a bandwidth adaptive feature sharing network was to have individual feature extraction networks with different components which project BEV maps to different deep feature spaces. However, because of the massive memory usage of this method and the long training process, this approach was canceled and altered with the idea of having a bank of dissimilar encoder/decoder pairs. Yet, in OPV2V, this idea has been implemented, and different large networks are structured and trained to reach specific compression rates. Although in the comparison of reached average precision for various compression rates, there is a remarkable trade-off between the AP and compressed data size, even with the tightest compression rate (4096x), the performance drops marginally only around 3%. Consider that all reached AP for deep feature sharing outperforming the RIS and FIS methods. While by sharing the fully processed information, the amount of required transmitting data would reduce to 3 KB, its average precision is almost 9% less than the 4096x DFS method. The other expressive number in Figure 1.27 is the size of non-compressed extracted features, which is about 66.56 MB (almost 70 times more than RIS). Due to the [31] details, feature maps are collected in sparse matrices, and it is feasible to compress them by lossless encodings. For example, in [31], the original spatial feature could be compressed firmly and shrink from 72.1 MB to 1.3 MB by lossless encoding. Not

considering the proper source encoding and overlooking a proper communication system could be one of the problems of this paper that avoid making a good compression. However, assuming that in the Deep features sharing case with a 4096x compression rate, it is not further possible to compress the data more (even with entropy encoding), at the stated scenarios, the following evaluation shows a firm performance for the DFS method.

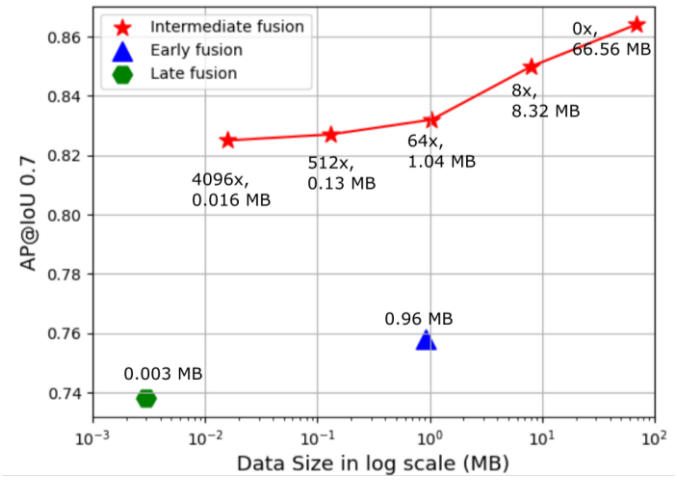


Figure 1.27 Average Precision with respect to data size [35]

1.4.8. Chapter Conclusion

After evaluating some of the most influential researchers that address the data transferring challenges of cooperative perception systems by proposing (unintentional!) semantic-based solutions, it appears it might not be any left area that deep learning-based available tools could modify. As we have seen in reviewed studies, due to their excellent comprehensive vision, their suggested solutions were not only limited to encoding/decoding the raw data, but they also cover the specific problems related to raw data projection to BEV space, recovery, aligning, and aggregation operation that must be done in the destination before final data fusion. Moreover, some of them also address the bandwidth limitations in their studies [31].

Let us return to the four significant functionalities highlighted in [7] that indicate the semantic-empowered network. These main features are semantic filtering, semantic preprocessing, semantic reception, and semantic control. The first three functionalities are employed in the V2V communication network by applying the deep learning-based proposed methods. However, for the fourth attribute, which could be a major one, we need some goal-oriented policy makings in these networks to have an agile arrangement of multimodal information gathering, fusion, and efficient resource utilization. In the next chapter, after analyzing the available and proposed strategies that are designed for cooperative perceptions communication. We would evaluate the potential to see how it would be possible to introduce some goal-oriented policy making.

2 Semantic-empowered Network Managing

2.1. The Available Strategies

Before proposing possible semantic-based strategies for V2V communications, it is worth mentioning some of the suggested standards and regulations being designed for this communication network. This review will help us first to see the developments achieved in these fields and second to get some idea about how other researchers look into V2V data transmission and to recognize their criteria for managing these communications.

The vital point needs some determined strategy and policy for cooperative vehicles, which have been addressed with vast numbers of research as well as proposals. European telecommunications standards institute (ETSI) established a technical report in 2019 [37] that proposes some standards and rules for generating collective perception messages (CPMs). Generally, A CPM is a message which contains technical information about the source vehicle, such as its onboard sensors (their range and field of view) and the information about detected objects such as position, size, and dynamic features (it seems mostly FIS type information). Based on the CPM structure, it particularly contains five containers; two of them are sensor information containers (SICs) and perceived object containers (POCs). Both containers are described as optional containers. SIC can report up to 128 sensors in a CPM, which is a section of transmitting vehicle embedded sensors capabilities that could exist in transmitting messages. The POCs can report up to 128 detected objects and the information about them, like their distance of coop vehicle, speed, dimensions, and, more importantly! The detection time.

There are also some regulations defining how frequently a vehicle should generate and transmit a CPM to other vehicles and which amount and types of information should be put in this message [38]. Based on the ETSI and authored rules concerning CPM generation, a vehicle must check every T_{GenCpm} that $10ms \leq T_{GenCpm} \leq 1000ms$. Besides this periodic CPM transmission, the vehicles should transmit a new CPM message for the four following conditions.

The first condition is about detecting a new object:

1. A vehicle should generate a new CPM if it has detected a new object.

The following conditions also apply for previously selected objects:

2. The absolute position of each detected object has changed by more than 4m since the last CPM message, which contains the position information of the corresponding object.
3. The absolute speed of each detected object has changed by more than 0.5m/s since the last CPM message, which contains the velocity information of the corresponding object.
4. This is more than 1 second from the last time the detected object was included in a CPM.

Each coop vehicle still generates a CPM every 1s, even if none of the above conditions are satisfied. Due to the ETSI regulations, the information about the onboard sensors must be included in the CPM only once per second. The ETSI proposed standards are basically the first set of rules which consider collective perception CPM generation. These rules are mostly considered as a benchmark for analyzing and investigating the performance of newly proposed methods [38].

The first optimization that can imagine for the CPM generation rules is to give a dynamic form to the T_{GenCpm}, and instead of transmitting CPM periodically, try to transmit it dynamically. [39] proposes a new dynamic CPM generation strategy and compares this dynamic strategy performance with the formulated and periodic strategy of established ETSI rules. As [39] explains, their new policy is based on varying the T_{GenCpm} based on the dynamics of the existing environment. As illustrated below, there is significant performance enhancement for dynamic policy compared to the regulated periodic policy.

The first comparison is with respect to the average CBR (channel busy ratio, which is measured by every second) factor. A high CBR means a more loaded channel and more risks of channel saturating.

Table 2.1 Averaged measured channel busy ratio (CBR) [39]

Policy	Traffic Density	CBR
Periodic at 2Hz	Low	5.6%
	High	11.9%
Periodic at 10Hz	Low	25.6%
	High	49.6%
Dynamic	Low	19.2%
	High	31.7%

The other experiment worth mentioning is Figure 2.1, which illustrates the detected object redundancy vs. the distance between object and vehicle. [39] determines the detected object redundancy factor as the number of updates received about the same object per second. There is a significant improvement in denying sending redundant information by using the dynamic strategy rather than the periodic 10Hz policy. (Which is for the default CPM generation T_{GenCpm} = 100ms)

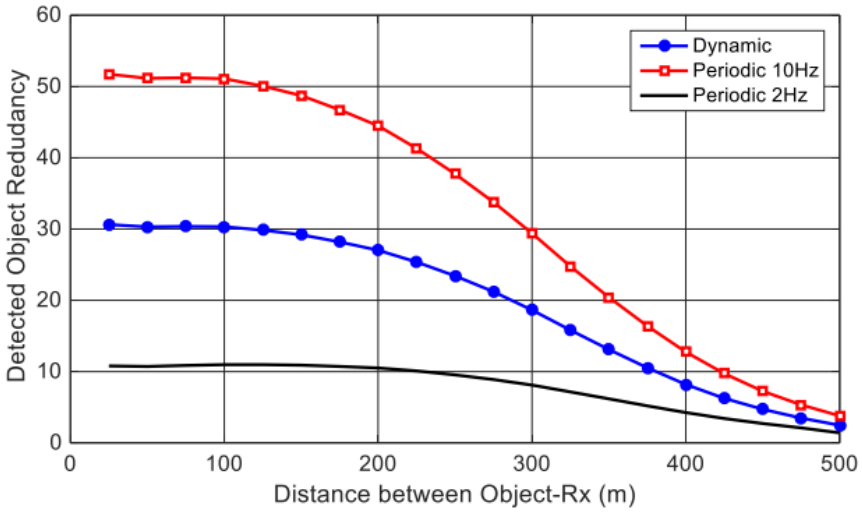


Figure 2.1 Detected object redundancy as a function of the object-rx distance [39]

In many V2V applications, the continuous exchange of broadcast messages is one of the critical services that should be provided. The absence of consistent mechanisms to verify the message delivery in broadcastings is one of the main challenges that [40] addresses and discusses. The first idea for solving this idea is to enable an acknowledgment system in all received vehicles. However, scalability in vehicular communications is one issue that makes it tough to implement an acknowledgment system, which means that a large number of surrounding vehicles in a broadcasting

case can cause simultaneous acknowledgments packet receiving at the sources side, which would lead to a declining in the reliability of acknowledgments delivery.

[40] proposes a context-based broadcast acknowledgment mechanism. At this mechanism, for each broadcasting action (and for particular broadcasted messages), some of the destination vehicles are deliberately requested to acknowledge receiving that critical data. Source vehicles will do retransmissions if they do not correctly receive the ACK. In addition, each specific V2V service must identify the conditions that activate the acknowledgment mechanism, and subsequently, the selected receivers must acknowledge the receiving that broadcast messages.

After trying to implement the idea of context-based broadcast acknowledgment by modifying the network layer in V2V communications, [40] also examines its proposed idea in a simulation environment. As this analysis results are illustrated below in Figure 2.2, the object detection probability improves by implementing the proposed mechanism. For example, when the proposed mechanism is set up for CounterReTx=3 (which means up to 3 chances for retransmissions of the CPM), the Object Awareness Ratio increases to 82% (about 30%~35% improvement) when the Object-Rx distance is about 50m.

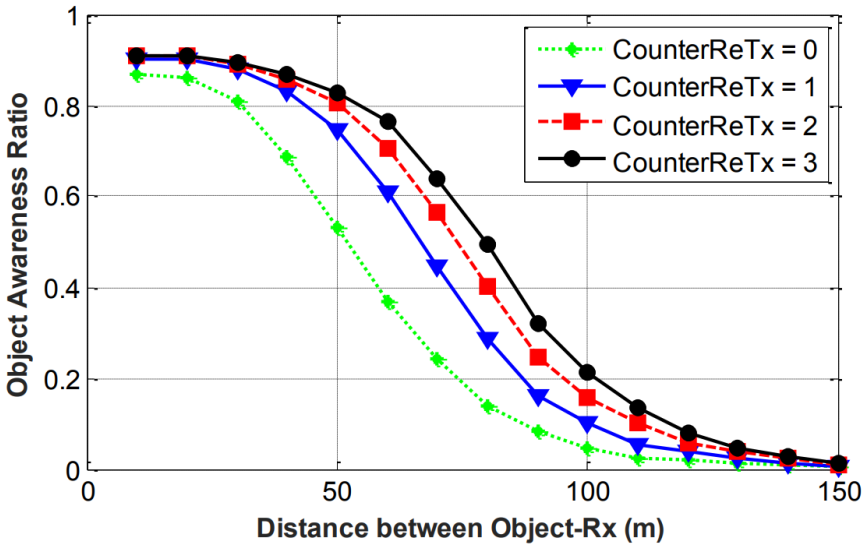


Figure 2.2 Object Awareness Ratio vs. the distance (The green line indicates the scenario with no ACK system). [40]

After all, the suggested approach can lower interference and collisions while also increasing accuracy.

Over and above the mentioned methods, another essential approach for regulating CPM messages is taking into account the value of CPM, which could be associated with the semantic attributes of the message. [41] is one of the attempts to make a value-anticipating network to deal with the massive data traffic in V2V communications.

The primary purpose of this study is to prioritize the information in a V2V network to enable efficient network resource allocations. The basic policy on selecting the content for sending in [41] is to have an awareness (and predictions) about the next CPM, and only if the observed value exceeds these anticipations for some pre-defined threshold would this content be elevated to be written in CPM. Moreover, this anticipation is being done by using three types of lists:

- Neighbor list
- CPM history for recent time window W
- (Their proposed idea) The anticipated CPM histories of remote vehicles. Which is a list of CPMs that adjacent vehicles are supposed to receive over the window W)

The novel tool for making a value-anticipating V2V communication in [41] is to obtain an anticipation about the CPM histories of neighboring vehicles in order to sort out the available information specifically for each of them. One of the underlying ideas for making this prior knowledge is this conservative assumption that each coop vehicle has only received a subset of CPMs of the ego vehicle's CPM history. After this assumption, the ego vehicle assigns a certain probability for each received CPM and each coop vehicle that demonstrates the CPMs obtainability for the coop vehicles.

The Following illustration, Figure 2.3 from [41], could explain the extreme case of this value anticipation. The vehicle v_0 has received two CPM messages and both are kept in its CPM history as $m_{0,1}, m_{1,2}$ ($m_{i,t}$: broadcasted CPM at timestep t_t, from vehicle v_i). Vehicle v_0 would anticipate a list (a subset of its own history list) of CPMs that exist in CPM histories of remote vehicles v_1 and v_2. This anticipation is based on the mentioned lists of v_0 (neighbor list and history) In this example, v_0 anticipation is that $m_{0,1}$ received by v_2 ($p_{0\to2}(t_1) > 0$), while $v_1$missed this CPM ($p_{0\to1}(t_1) = 0$). For $m_{1,2}$ that has been received by v_1 and is logged in v_0 history list, obviously, its receiving probability is equal to 1 for the $v_1$that has broadcasted it, $p_{1\to1}(t_2) = 1$ and v_1 knows $m_{1,2}$ is broadcasted. So, v_0 would put $m_{1,2}$ in the anticipated CPM history of v_1. While the $m_{1,2}$ CPM would be added to v_2 anticipated CPM history only with the probability $p_{1\to1}(t_2)$ that needs to be calculated by v_0.

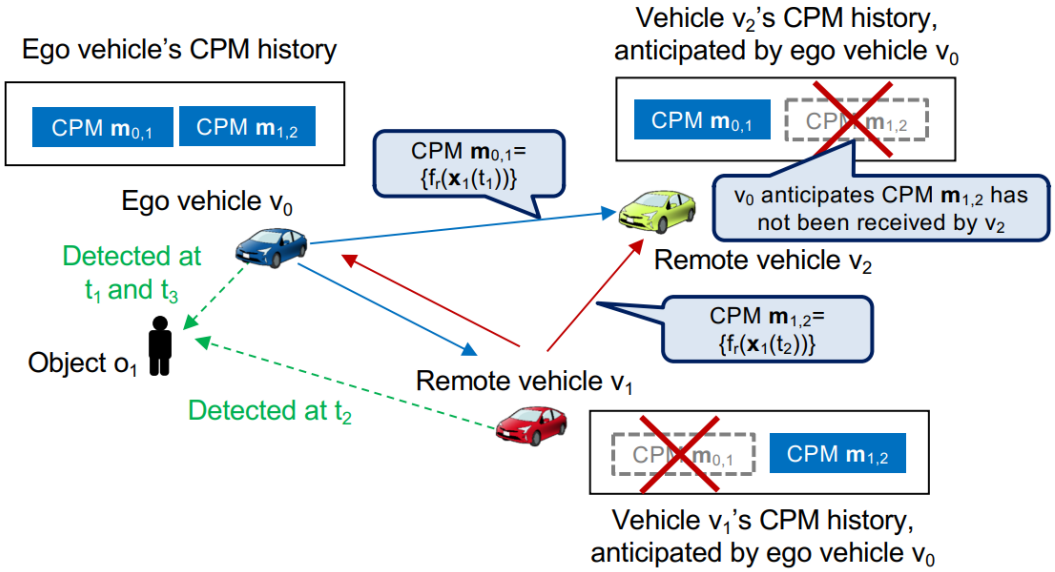


Figure 2.3 The Illustration of the explained straightforward example of prior knowledge anticipation for v_0

Although the proposed [41] could improve the V2V communication performance, it could only make prior knowledge about the obtained CPMs in other vehicles and does not care about the information particular importance for the destined coop vehicles.

One of the inspirations for creating a strategy for selecting higher-valued data with most perceived requests from destined vehicles is to leverage deep learning tools as a strategian which selects proper data for transmission with respect to mitigating the network potential load and enhancing communication reliability. [42] is one of the studies that try to implement deep reinforcement learning (DRL) to make a transmitting data gathering strategy, and it claims, compared to the baseline ETSI protocol, it was able to increase the detection accuracy by up to 12%. Their proposed DRL model is a deep Q-learning that benefits from a convolutional neural network to train their neural networks to maximize the long-term rewards. The states, actions, and rewards of this deep Q-learning are defined as follows:

- **States:**
  State s_t is made two information:
  - Circular projection
  - Network congestion level

  Circular projection, as shown below in Figure 2.4, is a projection that splits the vehicle's Field of View (FoV) into 5 × 3 grids. Then, each of these grids would be labeled with a category, which is written down in the following Table 2.2, that the label determines the grid status concerning four factors:
  1 --Local perception,
  2 --BSMs (Basic Safety Messages) transmission
  3 --CPMs (Cooperative Perception Messages) transmission
  4 --Object described by received CPMs.

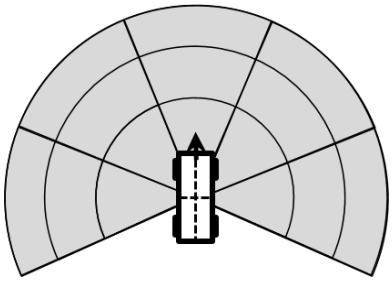


Figure 2.4 Field of view Circular Projection [42]

Table 2.2 Proposed categories for circular projection [42].

Category ID	Local Perception	BSM Transmission	CPM Transmission	Object from CPM
1	Empty	✗	✗	✗
2	Occupied	✗	✗	✗
3		✗	✗	✓
4		✓	✗	✗
5		✓	✗	✓
6		✓	✓	✗
7		✓	✓	✓
8	Occluded	✗	✗	✗
9		✗	✗	✓
10		✓	✗	✗
11		✓	✗	✓
12		✓	✓	✗
13		✓	✓	✓

The second part of the state, which is network congestion level and is showed by ψ, is calculated from the number of BSM and CPM messages received during the latest time window W. The network load is represented in 5 levels, varies from 1 (there are no surrounding vehicles: ψ=1) to 5 (vehicle density is as high as a congested urban area: ψ=5). This system also has this simplified assumption that the agent and the receiver are within the communication range, which means they have similar network conditions.

- **Actions:**

  The actions, which are determined by the Q-values generated by CNN, are the straightforward signal that decides transmitting (or not transmitting) CPM. Therefore, the action is from the action space {Transmit, Discard} corresponding to each CPM.

- **Reward:**
  The reward is designed to implement the main objects of this training, which are lowering the duplicated information in CPMs and improving the sensing capabilities. The reward Equation 2.1 has 4 terms, 1 reward and 3 penalties. Moreover, it is shown by $r_{t,\omega,\alpha,\beta}$ notation for time t, target object ω, transmitter α and the receiver β.
  More specifically:

$$r_{t,\omega,\alpha,\beta} = \lambda_{local} + \mu_{CPM} \times \theta_{t,\omega} + \mu_{hist} \times \phi + \mu_{netcong} \times C_{t,\beta} \qquad \text{Equation 2.1}$$

  - λ_{local} is a binary reward, that becomes 1 when the receiver has not detected the object ω.
  - Three μ (μ_{CPM}, μ_{hist}, and $\mu_{netcong}$) values are negative constants which are describing the penalties term.
  - $\theta_{t,\omega}$ represents the number of CPMs at time t that contain the information about object ω, which means more penalty for the same information that is shared by the multiple vehicles.
  - The term of μ_{hist} and ϕ shows the penalty that taking care of detection freshness. More recent timestamp to detect object ω leads to less penalty.
  - $C_{t,\beta}$ shows the network congestion level at time t for transmitting CPM toward the receiver β.

After explaining the different parts of this neural network, [42] evaluates the cooperative perception proposed strategy concerning the network load and packet reception ratio. The baseline for comparison of the gained benefits is the 10Hz constant CPM broadcasting. Each connected vehicle continuously transmits its CPMs for 100ms or whenever they detect new objects. As shown in the Figure 2.5, the proposed protocol hugely improves the amount of shared data. This performance also decreases the amount of shared data associated with the scene vehicle density.

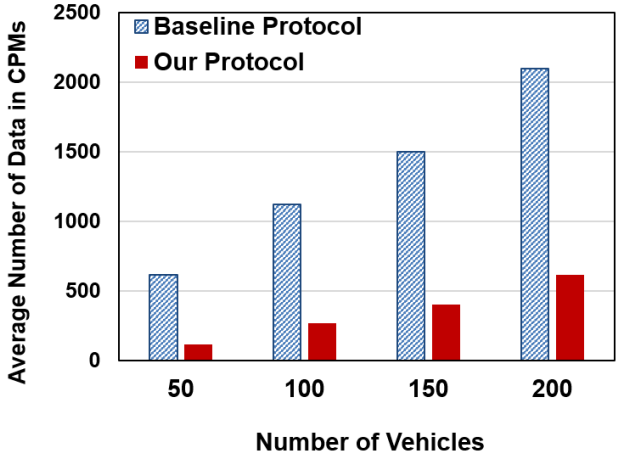


Figure 2.5 Average number of data shared in CPM with respect to the vehicles' density [42]

During the offline training process, [42] also monitors the neural network performance in terms of average detection ratio and packet reception ratio. Based on the illustration of Figure 2.6 and Figure 2.7, the network performance improvement during the training, this protocol increases the detection accuracy up to 12%, and the reception ratio is also increased to 27%.

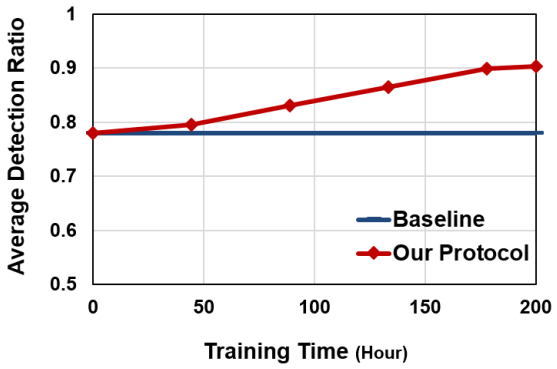


Figure 2.6 Average object detection enhancement during training the DNN [42]

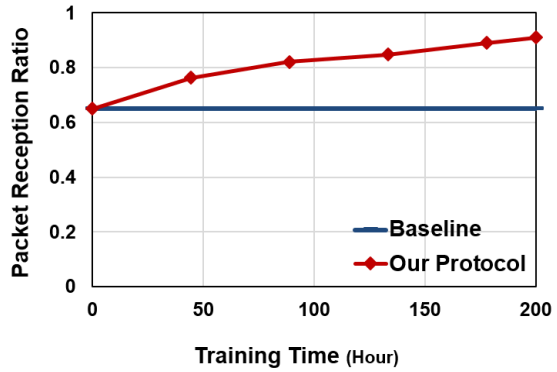


Figure 2.7 Average packet reception enhancement during training the DNN [42]

Let us take into account the different terms of reward function in the mentioned deep reinforcement learning method These terms implicitly reveal the innate trade-off between the amount of shared comprehensive spatial data (λ_{local}) and the freshness of that shared data ($\mu_{hist} \times \phi$). This trade-off also subjects to the networks' constraints ($\mu_{netcong} \times C_{t,\beta}$).

2.2. Goal-oriented Network Managing Strategies for V2V Communication Systems

Although by doing the former studies around the proposed cooperation perception strategies, we faced more sophisticated policies than ETSI [38] [40] [41], [43], we could not plainly call them semantic-empowered or goal-oriented network managing strategies. Implementing a context-based acknowledging system [40] or creating a history for the existing entities in a network based on gained prior knowledge [41] are general approaches for developing communication networks conventionally with no

regard to the primary purpose of communication. None of the reviewed studies leverage the main goal of communications as an advantage for improving the network.

Speaking about the primary purpose of cooperative perception communications, in this thesis, we decided to specify the main goal of this data exchange as promising clearer and expanded insights for vehicles in order to avoid any possible collisions and probable accidents in the environment. Therefore, ***collision avoiding*** and ***decreasing accident rates*** have been selected as the main goals. Routing and other applications could stand as less crucial purposes. Subsequently, we focus on the challenge of how the ego vehicle may use its gathered data about the environment to detect upcoming collisions and therefore manage its data exchanging for these predictions. In the rest of this session, we pursue for proposed approaches to predict future accidents (or behavior) in a traffic scene. Due to the innate complexity of traffic environments, most learned methods utilize deep learning-based pattern recognitions to predict upcoming incidents accurately. Some of these studies are presented and evaluated in the following:

2.2.1. Predicting Vehicles Future Movement in Dynamic Environments

2.2.1.1. Utilizing the Deep Neural Networks for Forecasting Interactive Actions

Most developed techniques for forecasting the other vehicles' (and drivers') behavior in different environments are associated with broader projects which address the challenges of self-driving vehicles. Almost certainly, one of the main challenges for a self-driving car is to share some driving spaces with human drivers who have very varied types of maneuvering and steering during driving. In more dense environments, this maneuvering is also considerably dependent on the behaviors of nearby cars and obstacles.

Understanding and predicting human drivers' intentions in a complex environment with considering all existing interactions with other objects could be complex for self-driving cars. Doing this task reliably requires knowing the environment and its available road topologies, tracking the drivers' motion histories and taking into account their past decisions, and analyzing each agent's interplay(s) with other agents and objects correctly and accurately. To give some examples of aged research that address this prediction challenge, we can mention quickly to following papers:

- **IntentNet: Learning to Predict Intention from Raw Sensor Data**
  The study [44] tries to develop a detector and forecaster that utilizes 3D point clouds produced by LiDAR captured data and dynamic maps of the environment. This project is based on a deep neural network (fully convolutional neural network, more specifically) that reasons vehicles' long-term trajectories. Particularly, for ego vehicle LiDAR extracted BEV map and road rasterized map as network's inputs, IntentNet is trained to generate the wanted three types of outputs.

1. Vehicle and background detection scores
2. Future action probabilities corresponding to the discrete intention
3. The current and predicted bounding box regressions for existing vehicles and objects.

The input and the general architecture of IntentNet are figured below in Figure 2.8 and Figure 2.9

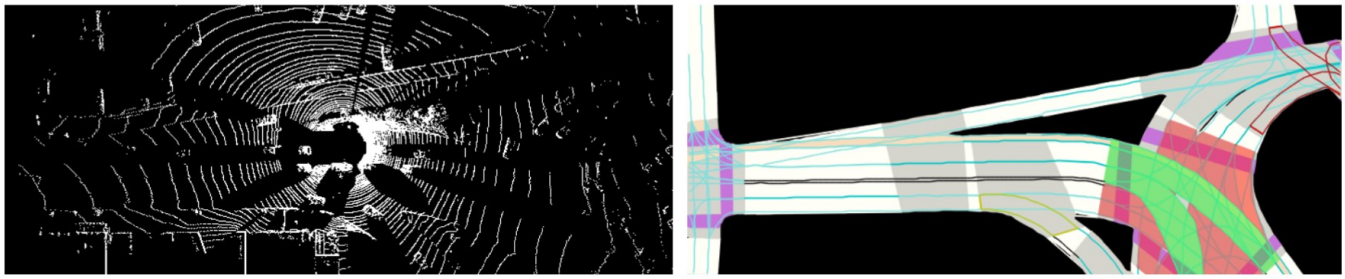


Figure 2.8 IntentNet inputs (Left: Voxelized LiDAR in BEV, Right: Rasterized map) [44]

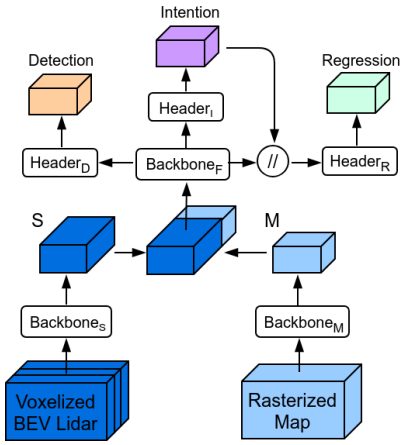


Figure 2.9 The overview of general IntentNet architecture [44]

- **End-to-end Interpretable Neural Motion Planner:**
  Another proposed neural motion planner for learning autonomous driving in complex situations is proposed in [45]. This neural network gets the input data, LiDAR raw captured data and an HD map, (similar to input variables of IntentNet) and generates the future trajectories of existing objects. This generation considers some essential elements of a complex urban area, such as traffic lights and interactions with other road users. Based on this prediction, this network also produces a cost volume that shows the goodness of potential positions that the ego vehicle can reach there by driving within. As this network name indicated, this network training is end-to-end with a multi-task objective function. The planning loss promotes the minimum cost plan like the trajectory performed by the human. An overview of this network has been displayed below (Figure 2.10). Please note that the Trajectory Sampler is a component that generates a set of possible

trajectories, then evaluates them with respect to a cost function. Among generated ones, the trajectory with minimum cost would be chosen.

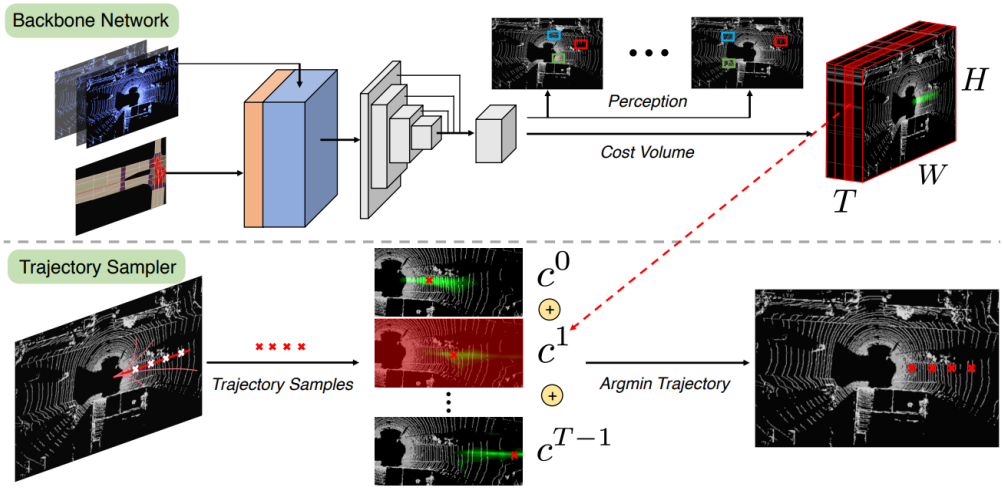


Figure 2.10 An overview of proposed interpretable neural motion planner [45]

In both earlier research based on using CNN networks for tracking and predicting the present vehicles, there was not any specific addressing to the issue of agents' interactions in the scene. Meaning that it seems these networks do not take into account the amount of each agent's awareness and watchfulness of its nearby objects. In the Intuitive human driving manner, we can assume two main principles:

- **Avoiding any kind of collision**
- **Staying on a straight path with a sluggishness for path swerving.**

Most of the proposed prediction system does not consider collision avoidance a significant factor in defining the object function. Due to their input formats, most training is only based on the vehicles' past trajectories and the current environments' states. However, we have found recent studies that propose DNN networks that predict future behavior by considering the dependencies between the traffic scene agents':

- **CAR-Net: Clairvoyant Attentive Recurrent Network:**
  One of the studies that also claims to exploit dependencies between agents' behaviors for path predictions is [46]. Still, the sources are similar to past studies: the past motion trajectory and a comprehensive, top-view scene image. The paper argues that realizing the complex interactions between agents and the environment requires knowing the scene context of that state. In order to extract the context of each scene, this paper suggests a deep attention-based model, which is named CAR-Net. The first module of this network is a feature extractor to compute feature vectors from the input (raw scene images). Then, the next module (visual attention module) derives context vectors of the input scene representing the image's most noticeable or essential areas.

Regarding attention-based models, These models could be generally categorized into single and multi-source attention models. The single source attention model extracts the features from a single image area. On the other hand, the multi-source attention model extracts the features from a combination of feature vectors distributed in all various parts of the image. The Car-Net applies both of these categories to obtain predictions on account of a broader spectrum of agent-space dependencies.

In order to highlight the differences between single and multi-source attention models, the following example could be helpful. For the beneath scene (Figure 2.11,Figure 2.12, and Figure 2.13) that is taken from an F1 circuit, the single source attention only can focus on a specific part of the scene, which in this example, it could be the coming turn. While multi-source attention can combine the two different spots of the circuit (coming and passing turns) simultaneously and see a broader complexity in the scene. These detected complexities and the history of observed trajectories would be used to predict the most likely path.

Figure 2.11 The highlighted areas obtained by single source component [46]

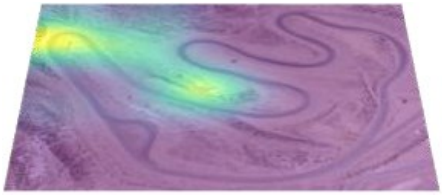


Figure 2.12 The highlighted areas obtained by mutli-source attention component [46]

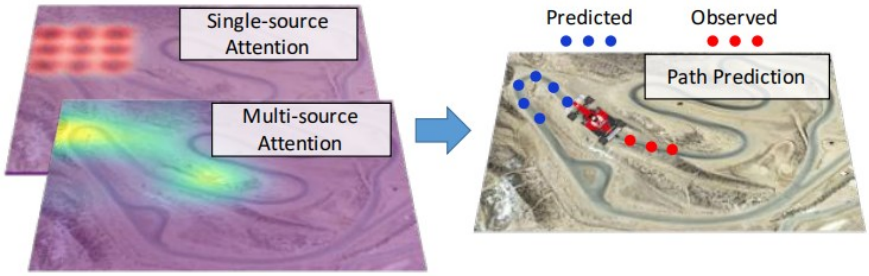


Figure 2.13 Predicting the future path by combining two attention mechanisms and using past trajectories histories [46]

The overview of CAR-Net architecture (consisting of three main modules which the feature extractor and attention are the first and second ones, respectively) is figured below in Figure 2.14:

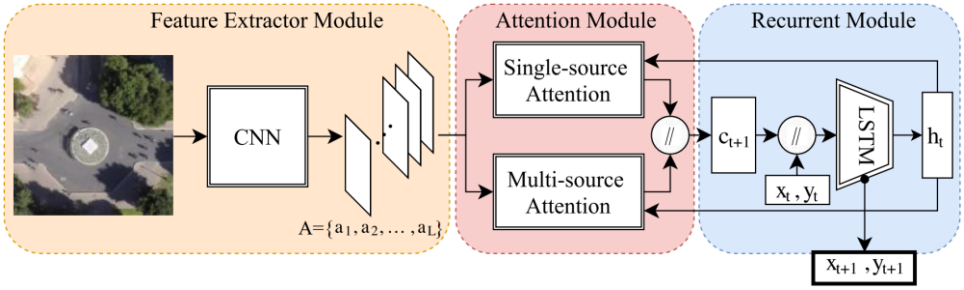


Figure 2.14 Overview of the CAR-Net architecture [46]

Implementing an attention module to detect the complexities in a map to perceive the environment more appropriately would give more precise predictions of the future. However, still, these source attention components could not entirely model the interactions of all agents, and if we direct our attention toward the nature of mobile vehicles, just knowing the shape of the scene and past vehicles' trajectory might not be complete enough data for evaluating the formations of all possible interactions.

One of the main studies that shape their research concerning interaction complexities is [47] which considers object interaction as graph-structured data and proposes a powerful model for processing this graph-structured data and forecasting the object's behavior. As we will see, the fundamental idea of this article, to look at a traffic scene as a graph of mutual interaction to predict behaviors, ***would stay as a key concept of our proposed idea.***

2.2.2. SpAGNN; Relational Behavior Forecasting Network

This network is presented in [47]. It is a spatially aware graph neural network, inspired by the Gaussian belief propagation method, that tries to model the existing interactions in a traffic scene to predict their coming states statistically. The general outline for modeling the available agents' interaction is to utilize a Graph neural network (GNN) which, while learning the system's dynamics by observing the past data, also tries to infer mutual interactions between available agents explicitly. In the following paragraphs, we will explain the GNN and its advantage for processing the graph-structured data in more detail. But before that, there are some descriptions of the SpAGNN main components:

1. **Object Detection Stage:**
   Like previous networks, the inputs of this network are the raw 3D LiDAR captured data and a raster HD map. The input map contains critical information about roads, lanes, intersections, traffic signs, and traffic lights. In order to ease the training procedure for the coming CNN networks, this information is encoded in separate

channels. Totally 17 binary channels are used to represent this semantic information.

The backbone of the detection component is based on a modified two-stream PIXOR network ([48] a proposed fully convolutional neural network for the real-time 3D object detection targets.), which is a lightweight object detection that its first stream is devoted to processes LiDAR Raw input, and the second one is for processing feed in HD maps. After extracting both maps' features, these mined features would be concatenated along the channel dimensions, which would be defined by the number of HD map available channels. Then, using a header CNN, these concatenated features are fused to make the input variable of two separated convolutional layers to determine the bounding boxes and corresponding confidence score of each anchor location. Furthermore, at the last level of the object detection state, there is a module that eliminates poor detections by applying a non-maximum suppression function on detected boxes. Eventually, the output of the detection stage is the set of bounding boxes and confidence scores of corresponding detected vehicles. The isolated overview of the object detection stage is figured in Figure 2.15.

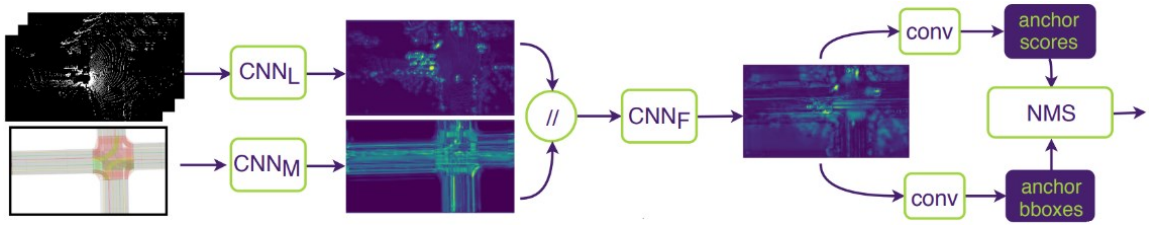


Figure 2.15 The overview of SpAGNN object detection stage [47]

2. **Interaction-based Behavior Prediction Stage:**

After detecting the vehicles dispersed in a traffic scene, the next level of the neural network is employed to predict the future states of agents statistically. The major key to this prediction is leveraging the interactions and relations between different actors to generate more accurate forecasts. The determined state in this framework is presented as $S_{i,t} = (x_{i,t}, \theta_{i,t})$ which $x_{i,t}$ is 2D waypoints and heading angles $\theta_{i,t}$ of ith agent at time t. [47] assumes a limited number of vehicles in each scene; based on this assumption, it suggests composing a fully connected directed graph to model agents' interactions in each scene. The main task of introduced graph neural networks (GNN) in the prediction stage is to figure out the importance and weights of the bidirectional defined interactions for each pair of agents. This study developed its spatial statistics methodology inspired by the Gaussian-Markov random field model (Gaussian MRF) and the Gaussian belief propagation (GBP), which is used to perform inference on graphical models with Gaussian distributions.

Since we will talk about GBP in more detail later, we will explain it briefly right now to understand the inspiring aspects of this model and find a background to continue the SpAGNN analysis.

2.2.2.1. Gaussian Markov Random Field

Based on the definition of a Markov network, this field (in the probability domain) is a set of random variables with Markov property which usually is displayed as an undirected graph. The concept of Markov property in a graphical model could be demonstrated in this way that in each MRF graph, for the node n, and its associated random variables X_n, and the set of neighbors $\mathcal{N}_n$ and all the nodes $\mathcal{N}$, the conditional probability of specific node n random variables could be write as Equation 2.2:

$$P(X_n|X_{\mathcal{N}} - X_n) = P\left(X_n\middle|X_{\mathcal{N}_n}\right) \qquad \text{Equation 2.2}$$

This formula fully characterizes the Markov property in an undirect graph. The graphical illustration of the Markov property in Figure 2.16 is that the black node is mathematically not subjected to other nodes by knowing the grey nodes. In other words, for given gray nodes, the black node would be independent of all non-neighbor nodes.

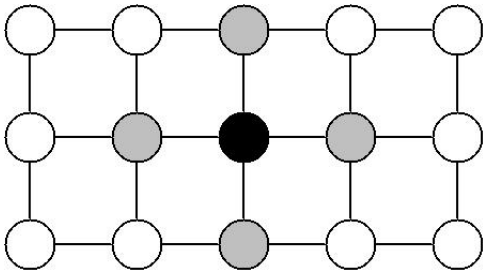


Figure 2.16 Due to Markov property, $P(X_{Black}|X_{All} - X_{Black}) = P\left(X_{Black}\middle|X_{Gray}\right)$

Since in SpAGNN, it is assumed that all the vehicles potentially can influence each other, and their modeled graph is fully connected, all of them make a clique, which is a subset of variables that each one is connected to all others. Assuming that the spatial state of all agents could be modelized as multivariate Gaussian distributions, for the observed status (that in SpAGNN is the LiDAR input data and HD map) symbolized by Ω, the conditional states probability distribution for given Ω could be re-evaluated in an exponential form:

$$p(s_1, \dots, s_N|\Omega) \propto \exp(\boldsymbol{s}^\intercal \boldsymbol{A}\boldsymbol{s} + b^\intercal \boldsymbol{s}) \qquad \text{Equation 2.3}$$

Where $\boldsymbol{A}, b$ are the parameters that model the existing interactions in the scene. Subsequently, these modeling parameters entirely depend on the Input Ω which determines the traffic scene. Additionally, this joint probability for the given observation can be decomposed to unary and pairwise potentials functions, which could be written in the following Canonical form (Equation 2.4).

$$p(s_1, \dots, s_N|\Omega) \propto \prod_i \phi_i(s_i, \Omega) \prod_{i,j} \psi_{ij}(s_i, s_j, \Omega) \quad \text{Equation 2.4}$$

Where $\phi_i(s_i, \Omega)$ is the unary potential defined as:

$$\phi_i(s_i, \Omega) = \exp\left(-\frac{1}{2} s_i^\intercal A_{ii} s_i + b_i^\intercal s_i\right) \quad \text{Equation 2.5}$$

And $\psi_{ij}(s_i, s_j, \Omega)$ is the pairwise potential:

$$\psi_{ij}(s_i, s_j, \Omega) = \exp\left(-\frac{1}{2} s_i^\intercal A_{ij} s_j\right) \quad \text{Equation 2.6}$$

Besides the above the Canonical presented form, the unary potential can be presented in Gaussian (the Moments) form as well:

$$\phi_i(s_i, \Omega) = \exp\left(-\frac{1}{2}(s_i - A_{ii}^{-1} b_i)^\intercal A_{ii} (s_i - A_{ii}^{-1} b_i)\right) \propto \mathcal{N}(s_i | A_{ii}^{-1} b_i, A_{ii}^{-1}) \quad \text{Equation 2.7}$$

Subsequently, based on the observations and the available model of cars' interactions, it is possible to apply an iterative belief propagation to obtain the marginal PDF of all existing variable data. The advantage of modeling the scene based on Gaussian distribution is that all the marginal PDF calculations (that potentially could be hard and resource-demanding ones) would be converted to simple sum-product processes.

The main core of Gaussian belief propagation, as its name points out! Is based on propagating the messages among all connected nodes. In Gaussian space, the messages also have the Gaussian form, so the generated messages (from node i_{th} to node j_{th}) could only be presented as a mean and a precision (in the Canonical form, or inverse variance in the Moments form) μ_{ij}, P_{ij} respectively. Therefore, the iterative equations (Equation 2.8 and Equation 2.9) for computing the messages would be: ($\mathcal{N}(i)$: the neighborhood set for node i_{th}, $\mathcal{N}(i) \backslash j$: same set without node j_{th})

$$P_{ij} = -A_{ij}^{-1} \left(A_{ii} + \sum_{k \in \mathcal{N}(i) \backslash J} P_{ki}\right) A_{ij}^{-1} \quad \text{Equation 2.8}$$

$$\mu_{ij} = -P_{ij}^{-1} A_{ij} \left(A_{ii} + \sum_{k \in \mathcal{N}(i) \backslash J} P_{ki}\right)^{-1} \left(b_i + \sum_{k \in \mathcal{N}(i) \backslash J} P_{ki}\, \mu_{ki}\right) \quad \text{Equation 2.9}$$

These iterative message passing would be continued till variable converging happens. Then, it would be time to apply the belief update step, which, thanks to Gaussian distribution, is simplified to established sums and products (Equation 2.10 and Equation 2.11).

$$P_i = A_{ii} + \sum_{k \in \mathcal{N}(i)} P_{ki} \qquad \text{Equation 2.10}$$

$$\mu_i = P_i^{-1}\left(b_i + \sum_{k \in \mathcal{N}(i)} P_{ki}\,\mu_{ki}\right) \qquad \text{Equation 2.11}$$

And lastly, the marginal probability for agent i_{th}:

$$p(s_i|\mu_i) = \mathcal{N}(s_i|\mu_i, P_i^{-1}) \qquad \text{Equation 2.12}$$

Although [47] claims that Gaussian Markov random field and implementing a correspondingly Gaussian belief propagation method could stand as proper models to predict agents' behavior by parametrizing their interactions as a linear equation, since it is not so accurate to define the state of an agent by some Gaussian distributions (for example, heading angles has a Von Mises distribution rather than Gaussian), They preferred to get some inspiration from Gaussian belief propagation to develop their technology which is founded on a graph neural network; that graph is being updated by performing message passing algorithms over the graph, similar to Gaussian belief propagation.

2.2.2.2. The Graph Neural Network Model

graph neural network model (GNN) first has proposed in [49] to extends existing neural network methods for processing the data presented in terms of graphs. At this study, to be able to apply the existing neural network approaches on an arbitrary graph $\mathcal{G}$ (with several possible types, such as acyclic, cyclic, directed, or undirected), a mapping function has been utilized to map all nodes of graph $\mathcal{G}$ (showing it with notation n) to a Euclidean space with m-dimensions: $\tau(G, n) \in \Re^m$.

Without diving deep into this inspecting this research, let us explain this methodology by taking a look at the following case (Figure 2.17) in point:

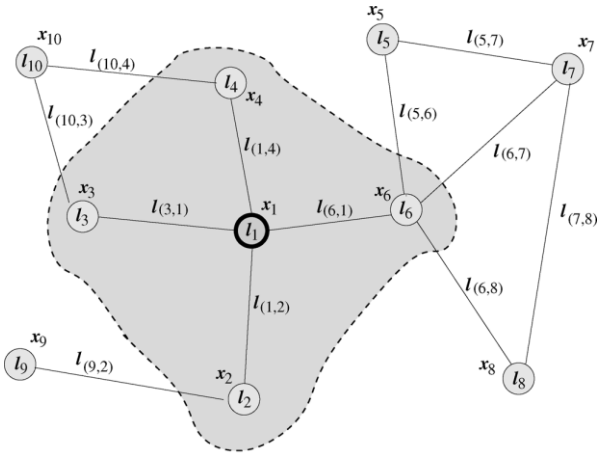


Figure 2.17 An Example of GNN from [49]

For the stated graph structure Figure 2.17, the general state of node 1, x_1,could be calculated with respect to the information have gotten from the neighbors of corresponding node:

$$x_1 = f_w\left(\iota_1, \iota_{(1,2)}, \iota_{(1,3)}, \iota_{(1,4)}, \iota_{(1,6)}, x_2, x_3, x_4, x_6, \iota_2, \iota_3, \iota_4, \iota_6\right) \quad \text{Equation 2.13}$$

Where f_w is a function, called local transition function, that represents each node dependency on its neighborhood. After calculating the node state x_1, to produce the corresponding output o_1, a local output function g_w is needed to make the desired output: $o_1 = g_w(x_1, \iota_1)$

Next, after formulating the dependencies in a graph and how to calculate the corresponding output in a given graph:

$$\boldsymbol{x}_n = f_{\boldsymbol{\omega}}\left(\iota_n, \iota_{co[n]}, \boldsymbol{x}_{ne[n]}, \iota_{ne[n]}\right) \quad \text{Equation 2.14.a}$$

$$\boldsymbol{o}_n = g_{\boldsymbol{\omega}}(\boldsymbol{x}_n, \iota_n) \quad \text{Equation 2.14.b}$$

It is time to apply the neural network model to this graph-based model, to learn the interactions, and find the local transition and output functions based on neural networks. At this point, it is required to encode the desired graph neural network to the equal classic (and unfolded) neural networks to start doing the training procedures to obtain local functions.

The [49] illustrates its idea by unfolding the simple graph (Figure 2.18) to a classic feedforward neural network.

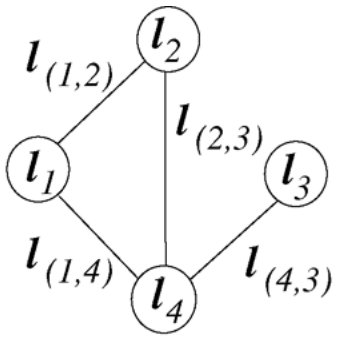


Figure 2.18 An arbitrary example graph of [49]

By considering the timestep t, and the local functions that ultimately extracts the output vector at time t corresponding to each node, this given graph can be re-arranged and re-structured to an encoding network (Figure 2.19). The main transformation of encoding networks is replacing the nodes with local functions, $f_{\boldsymbol{\omega}}$ and $g_{\boldsymbol{\omega}}$, where these functions are implemented by feedforward neural networks.

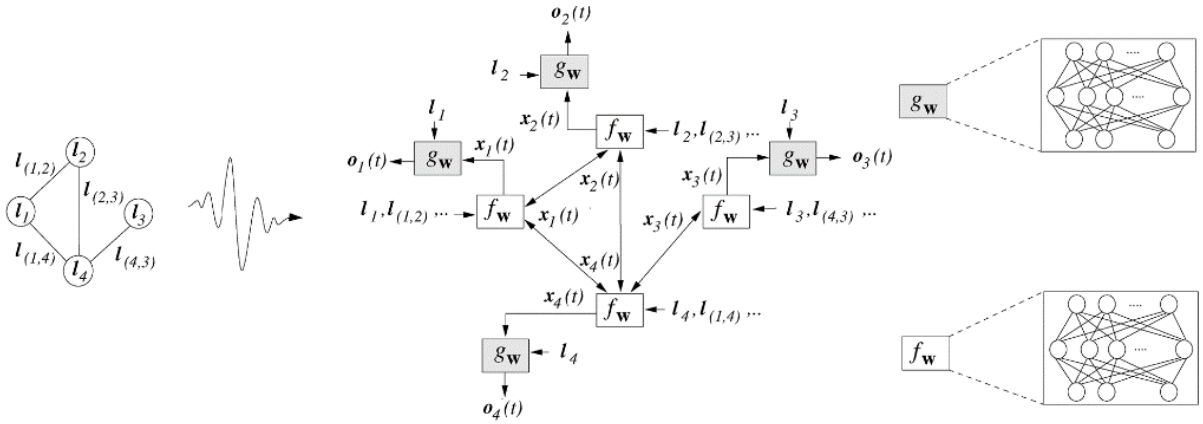


Figure 2.19 Illustrating the explained conversion from a graph to an encoding network [49]

The next step for implementing the training procedure on feedforward neural networks is to unfold the obtained encoding network and convert the inevitable and existing cycles to recurrent neural networks. The achieved unfolded network is time-dependent, and each layer relates to a corresponding time instant. After the unfolding procedure, the type of connections between timestep layers depends on the encoding network and the set of each node's neighbors.

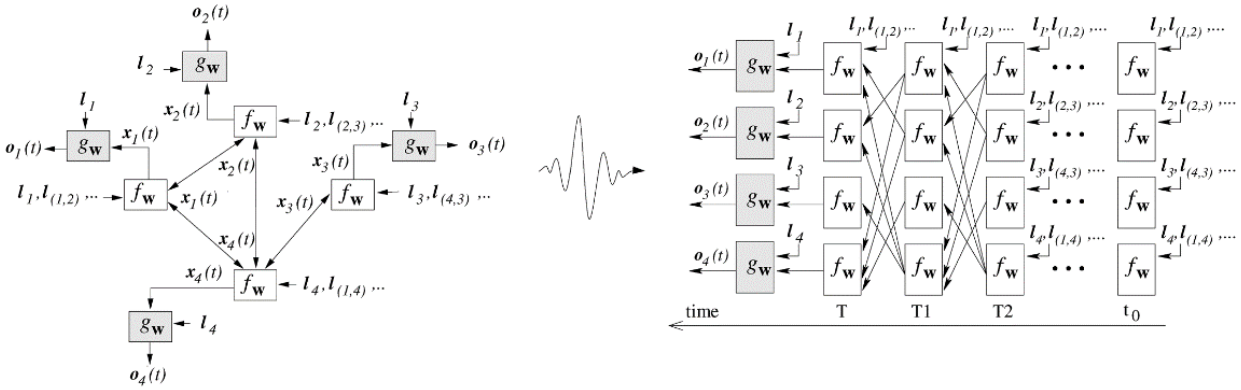


Figure 2.20 The explained conversion to break the cycles, and to unfold the encoding network to a recurrent neural network. [49]

[47] states two main reasons for using the GNN approach for predicting:

1. The model size does not depend on the input graph size.
2. The high capacity to learn decent representations at both graph and node levels.

In the interaction graph (which GNN would be utilized to resolve it), each node corresponds to an existing vehicle (or object) at the scene. Inspired by Gaussian MRF, the SpAGNN node state consists of two separate components: named hidden and output states. The hidden state which is represented as h_v is an extracted vector associated with region of interest (RoI) feature map for the v_{th} vehicle. After aligning and extracting a fixed-size spatial feature map for vth bounding box, this map then is being down-sampled and projected to a 1D feature vector by a CNN and max pooling

networks respectively. Figure 2.21 shows the convolutional neural network, CNN, which extracts the initial hidden state vectors by getting the aligned detected boxes as the input.

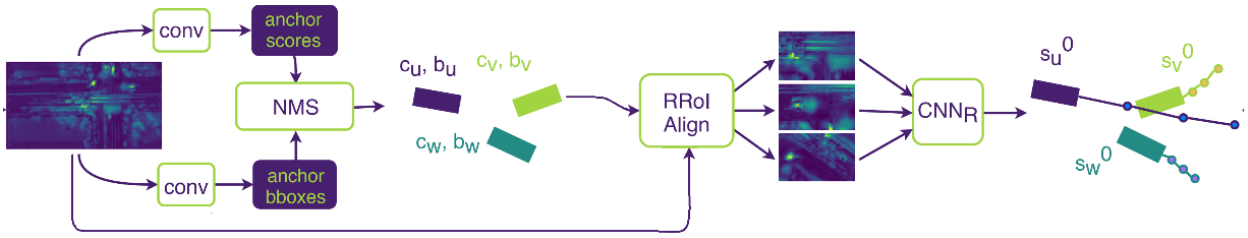


Figure 2.21 The initial level of the SpAGNN networks that extract the initial hidden state for all nodes and pass it to the SpAGNN network. [47]

Motivated from GBP, the output states of all nodes in the SpAGNN network consists of statistics of the marginal distribution. The two key parameters for describing the future behavior of agents are the waypoint and heading angles of vehicles, and the [47] chooses Gaussian and Von Mises probability distributions to model these two parameters respectively. Therefore, the marginal distribution of the node associated with v_{th} vehicle at k_{th} step of message passing would be:

$$p(\boldsymbol{x}_v^k|\Omega) = \mathcal{N}(\boldsymbol{x}_v^k|\boldsymbol{\mu}_v^k, \boldsymbol{\Sigma}_v^k)\ \&\ p(\boldsymbol{\theta}_v^k|\Omega) = \mathcal{N}(\boldsymbol{\theta}_v^k|\eta_v^k, \kappa_v^k) \qquad \text{Equation 2.15.a}$$

Where:

$$\boldsymbol{x}_v^k = [x_v^k, y_v^k]^\intercal \qquad \text{Equation2.15.b}$$

$$\boldsymbol{\mu}_v^k = \left[\mu_{x_v}^k, \mu_{y_v}^k\right]^\intercal \qquad \text{Equation 2.15.c}$$

$$\boldsymbol{\Sigma}_v^k = \begin{pmatrix} {\sigma_{x_v}^k}^2 & \rho_v^k \sigma_{x_v}^k \sigma_{y_v}^k \\ \rho_v^k \sigma_{x_v}^k \sigma_{y_v}^k & {\sigma_{y_v}^k}^2 \end{pmatrix} \qquad \text{Equation 2.15.d}$$

Therefore, the output state for each node at each iteration is a set of all needed parameters to describe the predicted waypoint and heading angles of the corresponding vehicle statistically. The main purpose of iterating each message passing is gradually improve the output states by converging to some optimum situations. The initialized output state for setting up the propagation process is obtained by utilizing a multi-layer perception model, which takes the hidden state of the node v ($h_v^{k=0}$) and gives a prediction of the initial output state.

The next challenge for developing a GNN inspired by Gaussian BP is to create a framework concerning composing and passing the generated messages at each node and then establishing a way to update the nodes' state with respect to the nodes'

received messages. For this matter, [47] proposes a further MLP feedforward neural network that generates the direct message from node i_{th} to node j_{th} at k_{th} timestep; $m_{i\to j}^{k}$.

The message is computed as:

$$m_{i\to j}^{k} = \varepsilon^{k}\left(h_i^{k-1}, h_j^{k-1}, \tau_{i,j}\left(o_i^{k-1}\right), o_j^{k-1}, b_i, b_j\right) \qquad \text{Equation 2.16}$$

Where the MLP input vectors (in Equation 2.16) are:

- h_i^{k-1}, h_j^{k-1}: the hidden state of both ith and j_{th} nodes at times-step k-1_{th}
- o_j^{k-1}: the output state of destination (j_{th} nodes) at times-step k-1_{th}
- $\tau_{i,j}\left(o_i^{k-1}\right)$: the output state of source node (i_{th} nodes) that is translated with respect to destination system (by $\tau_{i,j}(.)$ Transformation matrix that as [47] claims that this prior transformation will ease the training procedure significantly.)
- b_i, b_j: The detected boundary boxes corresponding to i_{th} and j_{th} vehicles.

After generating the messages, which are the output vectors of 3-layer MLP $\varepsilon^{k}(\cdot)$ neural network, and passing these messages to destination nodes, the aggregation of received messages (vectors) is done by an arithmetic operation. The selected operation in [47] is a feature-wise max operator. So, the aggregated message a_j^{k} is:

$$a_j^{k} = A\left(\left\{m_{i\to j}^{k} : i \in N(j)\right\}\right) \qquad \text{Equation 2.17}$$

Finally, after collecting and combining all messages, it is the time for updating each node state due to its received messages. This state updating requires to first update the hidden state of the j_{th} node $h_j^{k-1} \xrightarrow{Updating} h_j^{k}$, and this updating is being done by utilizing a Gated recurrent unit (GRU) cell, which simulates and assemble the long-term dependencies in iterative receiving messages. The inputs of this GRU cell are the aggregated message a_j^{k} and the past hidden state of the node h_j^{k-1}. This state updating could be demonstrated as: $h_j^{k} = \mathcal{U}^{k}\left(h_j^{k-1}, a_j^{k}\right)$. Then, as same as the initial output states, the output state for k_{th} timestep is obtained by utilizing a 2-layer MLP: $o_j^{k} = \mathcal{O}^{k}\left(h_j^{k}\right)$

Considering that the training process for SpAGNN is a jointly end-to-end method, and the object function corresponding to this joint training is a multi-task one. In terms of comparison of the SpAGNN, [47] makes a comprehensive comparison containing much possible interaction and motion forecasting methods or neural networks. These experiments have been done by the collected ATG4D dataset [50], and the investigations generally cover two main areas, joint detection and prediction, and social interaction and motion forecasting. [47] decides to use "precision vs. recall" for IoU=0.7 and collision rate vs. recall curves to evaluate detection and prediction performance. Figure 2.22 and Figure 2.23 show a meaningful improvement in precision vs. recall and a significant enhancement in collision avoidance by utilizing the SpAGNN method, which cooperatively verifies the joint detection and prediction performance.

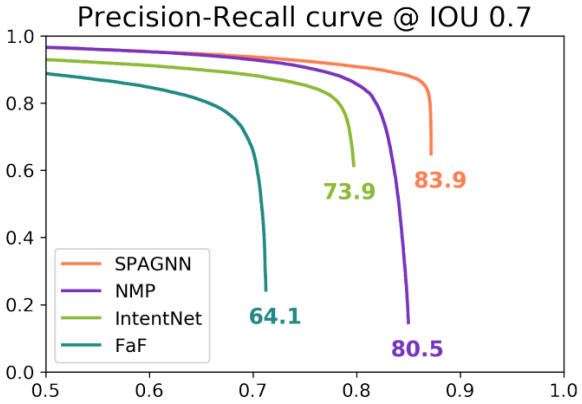


Figure 2.22. Precision vs Recall Comparison for different joint detection and prediction systems, based on ATG4D database [47] [50]

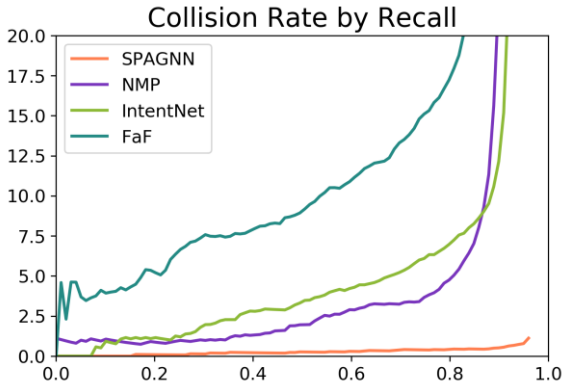


Figure 2.23. Collision Rate vs Recall Comparison for different joint detection and prediction systems, based on ATG4D database [47] [50]

In order to evaluate the social interaction motion forecasting that is resulted from the SpAGNN method, the [47]determines three different parameters. The first is the cumulative collision rate associated with vehicle interactions. The second and third ones are about motion forecasting, which are absolute Least square centroid and absolute heading errors. Table 2.3 illustrates these three indicators for different systems, and at 80% detection recall, the SpAGNN can outperform other existing systems and methods.

Table 2.3 A comprehensive comparison of different network model considering social interaction and motion forecasting metrics at 80% detection recall [47]. The table evidently shows SpAGNN outperforming.

Model	Cumulative Collision Rate %		absolute Least square centroid error (cm)			Absolute Heading Errors (deg)		
	0-1s	0-3s	0s	1s	3s	0s	1s	3s
D+T+S-LSTM [Ref]	1.43	16.31	22	147	607	4.06	5.14	8.07
D+T+CSP [Ref]	1.64	20.78	22	95	282	4.06	4.70	6.20
D+T+CAR-Net [Ref]	0.28	12.30	22	46	149	4.06	4.87	6.14
FaF [Ref]	1.12	17.41	30	54	183	4.71	4.98	6.43
IntentNet [Ref]	0.28	7.03	26	45	146	4.21	4.40	5.64
NMP [Ref]	0.05	3.06	23	36	114	4.10	4.24	5.09
E2E S-LSTM [Ref]	0.06	1.14	22	36	106	4.97	4.85	5.61
E2E CSP [Ref]	0.06	4.47	23	38	114	4.82	5.04	5.84
E2E CAR-Net [Ref]	0.07	1.15	22	35	105	4.44	4.41	5.12
SpAGNN	**0.03**	**0.42**	**22**	**33**	**96**	**3.92**	**3.89**	**4.55**

At the end it is worth spelling down the SpAGNN method in an algorithmic way (Algorithm 2.1) to summarize it, and more importantly, to see its similarities with classic belief propagations more clearly.

Algorithm 2.1: SpAGNN; Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting

Algorithm Input(s):

- $\Omega = \{\mathrm{L}, \mathrm{M}\}$ where L is LiDAR input and M is HD map
- Number of Iterations K

Algorithm Output(s):

- $Detections\ which\ consits\ of\ confidenc\ and\ bounding\ box\ parameters$:
  $\mathrm{D} = \{c_0, b_0, \dots, c_N, b_N\}$
- Output states corresponding to each detected vehicle
  $O^K = \{o_0^K, o_1^K, \dots, o_N^K\}$

1:	$\mathrm{Detection}_{\mathrm{Network}(\Omega)} \rightarrow \{c_0, b_0, \dots, c_N, b_N\}$	➢ Detecting the existing vehicles by utilizing object detection network.
2:	**for** $i = 1{:}N$ **do**	➢ For each detected vehicle, Compute initial features
3:	$\mathrm{RRoiAlign}(b_i) \rightarrow r_i$	➢ Handing the aligned detected boundary boxes to CNN network to extract the initial hidden state.
4:	$MaxPooling(CNN(r_i)) \rightarrow \mathrm{h}_i^0$	
5:	$\mathcal{O}^0(h_j^0) \rightarrow o_j^0$	➢ Utilizing MLP network to get initial output state from extracted hidden state
6:	Construct illustrative graph for detected object $G = (\mathrm{V}, \mathrm{E})$	
7:	**for** $i = 1{:}K$ **do**	➢ Repeating the message iterating K times
8:	**for** all $(i, j) \in E$ **do**	➢ Compute message for every edge combination in the graph
9:	$\varepsilon^k(h_i^{k-1}, h_j^{k-1}, \tau_{i,j}(o_i^{k-1}), o_j^{k-1}, b_i, b_j) \rightarrow m_{i\rightarrow j}^k$	
10:	**for** all $v \in V$ **do**	➢ After sharing all messages, it is time for all nodes belief updating
11:	$A(\{m_{i\rightarrow j}^k : i \in N(j)\}) \rightarrow a_j^k$	➢ Aggregating all the received messages
12:	$\mathcal{U}^k(h_j^{k-1}, a_j^k) \rightarrow h_j^k$	➢ Updating the node hidden state w.r.t to aggregated message
13:	$\mathcal{O}^k(h_j^k) \rightarrow o_j^k$	➢ Utilizing MLP network again to extract output state from updated hidden state
14:	**return** $D, \mathcal{O}^k$	

2.3. Graph-based Interaction Modeling

As SpAGNN inspired us, it seems that it could be feasible to rely on graph-based methods to detect the upcoming collisions in a dynamic traffic scene and predict the coming collisions in a different pattern of dispersed vehicles by considering the type of their mutual interactions. In other words, by projecting the dynamic properties of detected vehicles to nodes of an interaction graph and creating their edges by taking into account some mutual parameters (such as each vehicle's awareness of the other vehicles, the vehicles' perception of the surrounding environment, and the specific configuration of vehicles' sensors and their corresponding peripheral view) we could predict vehicles short-term behavior, and due to that, predict the coming collisions.

As we mentioned earlier, we can summarize the main human objects in momentary driving in these two:

- Avoiding collisions.
- Staying on a straight path with a sluggishness for path swerving.

Regarding these intentions and the newly obtained forecasting tool (very likely empowered by Deep learning methods), it seems that it could be possible to compose some new semantic-based frameworks that take care of information selection, landing vehicle(s) choosing, and transmission policies.

Moreover, a proper proposed framework could also play a key role in reducing the network load and improving the agents' cooperation to earn a broader perception. As we see in analyzed experiments [35], it is too important to have a specific strategy due to the significant volume of sensor data (especially at RIS) and the emphasis on sharing not-aged value. The non-strategic transmission would not only lead the network congestion and channel overburdening, but also since the data corresponding to vehicles' observations is so time-sensitive, the resource allocation for aged and outdated data could be pointless, or in inferior cases, it could end up in erroneous perception expansions.

Furthermore, it also must be considered that in the fusion process, it is so probable that aggregated data include vast amounts of overlapping fields of view. Although this observation data profusion can be employed effectively to increase the density of some spatial areas to lower false negative detection, at the end, there is an information redundancy that would lead to wasted communication (and even computing) resource allocations. Moreover, this undeliberate transmission is even less acceptable because, far apart from the redundant data sharing, this disarrangement could lead to not sharing some usable (or even essential) data as well, which would raise doubts about the collision pretending (the perception expanding procedure general goal).

3 Factor Graph-Based Collision Avoidance

As we discussed earlier in SpAGNN, belief propagation (BP) is an inference algorithm for evaluating the marginals of joint distributions by message passing. This message passing can be done in different forms of graphs. We have seen the formula and steps of the BP methods for the Markov random field earlier. Likewise, this approach can be utilized for factor graphs as well.

3.1. Factor Graph

A factor graph is a probabilistic graphical model consisting of two types of nodes: variable and factor types. Variables (in our field of study) could be interpreted as the statistical states be provided by observation or predictions. Factors are also nodes that define the relationships between variables in the graph.

In this kind of graph, the Hammersley-Clifford theorem declares that, for any clique, a set of variable nodes that each one is connected to all others, the global function $\phi(\mathrm{X})$ can be presented as a product of factors $\phi_i(X_i)$:

$$\phi(\mathrm{X}) = \prod_i \phi_i(X_i) \qquad \text{Equation 3.1}$$

That for ith factor node, X_i denotes the adjacent variable nodes that are connected to factor nodes ith, and subsequently, ϕ_i is the function assigned to this factor node. An example of this factorization is put below [51]:

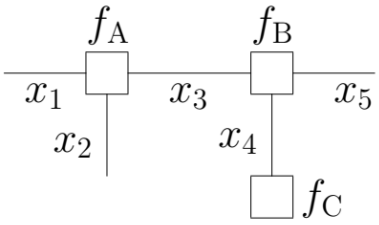


Figure 3.1 An Example of factorization done by factor nodes [51]

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3, x_4, x_5) \times f_B(x_3, x_4, x_5) \times f_C(x_4) \qquad \text{Equation 3.2}$$

One of the main applications (employed in Gaussian MRF analysis) is to calculate the joint distribution of some variable nodes' states. For example, for the below factor

graph that demonstrates a Markov chain [51], the joint distribution can be obtained by the Hammersley-Clifford theorem:

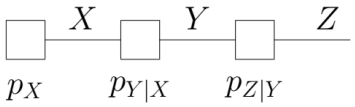


Figure 3.2 A Markov chain example [51]

$$p_X(x,y,z) = p_X(x) \times p_{Y|X}(y|z) \times p_{Z|Y}(z|y) \quad \text{Equation 3.3}$$

Since the Gaussian states are stated as some exponential terms, it is worth pointing out that one of the primary factor nodes explanations is energy-based models where each factor connecting the subset of Variable nodes X_{I}, defines an energy associated with the subset X_{i}:

$$f_i(X_{\mathrm{i}}) \propto e^{-E_i(X_{\mathrm{i}})} \quad \text{Equation 3.4}$$

The benefit of using an energy-based model is that founded X_{MAP} states would lead to minimizing the log of factor energies. Because:

$$\begin{aligned} X_{MAP} &= \arg min_X \left[-\log p(X)\right] = \arg min_X \left[-\log(\prod_i e^{-E_i(X_{\mathrm{i}})})\right] \\ &= \arg min_X \left(\sum_i E_i(X_{\mathrm{i}})\right) \end{aligned} \quad \text{Equation 3.5}$$

Besides the appropriate representation flexibility that could be obtained by using factor graphs, subject to their ability to represent joint probabilities in some factorization forms. Another benefit that comes (with representing system states by variable nodes, and PDF or energy function over a set of variables nodes by factor nodes) is to can maximize the posterior the predicted X_{k} stated given the history of measurements Z_{k} [52]:

$$\mathrm{X}_{\mathrm{k}}^{*} = \arg\max_{\mathrm{X_k}} \mathrm{p}(\mathrm{X_k}|\mathrm{Z_k}) \propto \mathrm{p}(\mathrm{X_0})\mathrm{p}(\mathrm{Z_k}|\mathrm{X_k}) \quad \text{Equation 3.6}$$

One of the main applications of factor graph characterization, and its maximization property, is in inference problems in the robotic fields [53]. More specifically, In SLAM (simultaneous localization and mapping) problems, the main focal point of these localization problems is to exploit the obtained observations Z to characterize the unknowns X, including robot poses and the unknown milestones [53].

By looking at these estimation problems from a probabilistic point of view, these problems can be described by Bayesian probability as some conditional density p(X|Z) (that is called probabilistic inference) and subsequently, estimating the X by maximum a posteriori or MAP estimation.

$$X^{MAP} = \arg\max_X p(X|Z) \qquad \text{Equation 3.7}$$

This type of description p(X|Z) is called probabilistic inference. By knowing the probabilistic model for the variables of interest, and the exploited structure of SLAM problem variables as the set of prerequisites, thanks to factor graph plotting, it would be possible to provide a mechanism to describe complex probability functions to find the maximum a posteriori estimation. (MAP).

Speaking of computing variables' marginal posterior distribution, we return to the belief propagation (BP) algorithm that could be implemented as an iterative message passing on factor graphs to update a node's posterior distributions by sending and receiving messages. In the domain of factor graphs, the belief propagation algorithm can be divided into three main phases: [54]

3.1.1. Factor Graph Message Passing

- **Factor-to-Variable Message Passing**:
  Includes sending a message from each factor node to all its adjacent connected variable nodes (Considering that factor nodes can only link to variable nodes and vice versa). The phase of sending a message to a specific variable node consists of the following actions:
  1. Aggregating all received messages from all other adjacent variable nodes.
  2. Marginalizing over the other nodes' variables to compose the desired message (called factor's belief over the receiving node).

  The graphical and mathematical illustrations are put below [54]:

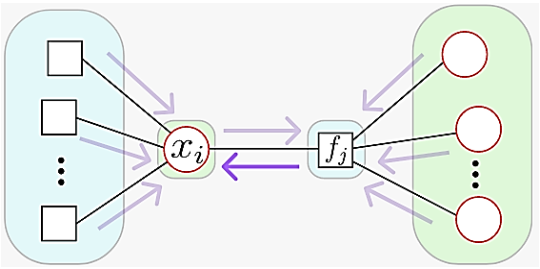


Figure 3.3 Factor j_{th} to variable i_{th} message passing

Different Steps Mathematical Expression:

i. **Messages Gathering:**
   It would be done by the repeated multiplication over all received messages:

$$\prod_{k \in N(j) \setminus i} m_{x_k \to f_j} \qquad \text{Equation 3.8}$$

ii. **Marginalization:**
After gathering the received messages, it is time to calculate the output of jth factor with respect to variables of adjacent variable nodes: $f_j(X_j)$, then compute the marginalization with respect to the variables of other variable nodes:

$$m_{f_j \to x_i} = \sum_{X_j \setminus x_i} \left(f_j(X_j) \times \prod_{k \in N(j) \setminus i} m_{x_k \to f_j} \right) \quad \text{Equation 3.9}$$

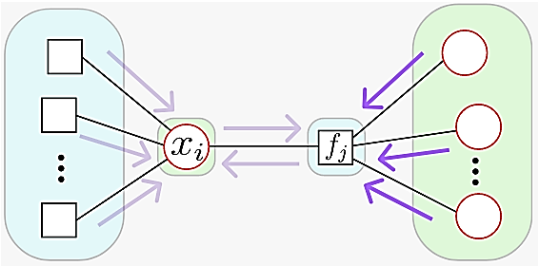


Figure 3.4 Aggregating all messages received from other variable nodes [54]

- **Variable-to-Factor Message Passing:**
Initially, [54] interprets the variable to factor message as the belief of the variable if the receiving factor node did not exist. This message is easily prepared by taking the product of all messages passed to the variable node from its connected factor nodes (excluding the receiving factor). Mathematically:

$$m_{x_i \to f_j} = \prod_{s \in N(i) \setminus j} m_{f_s \to x_i} \quad \text{Equation 3.10}$$

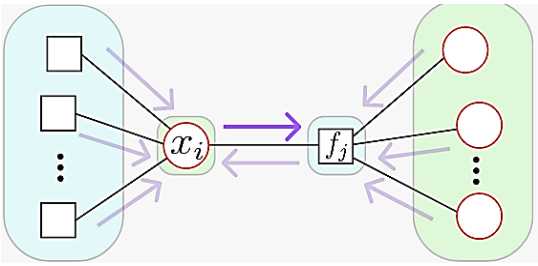


Figure 3.5 Variable ith to factor jth to message passing [54]

- **Belief Updating:**
The variable node state (that is called beliefs in [54]) is renewed by gathering all the messages that adjacent factor nodes have sent. The gathering process is taking the product of the incoming messages as it is demonstrated:

$$b_i(x_i) = \prod_{s \in N(i)} m_{f_s \to x_i} \quad \text{Equation 3.11}$$

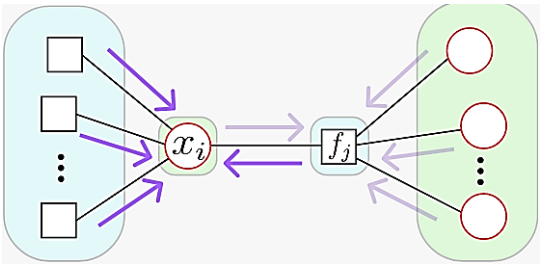


Figure 3.6 The variable nodes updating process would be done after receiving all the messages from adjacent factor nodes [54]

3.1.2. The Convergence for Belief Propagation Method

One of the main issues about belief propagation is ensuring the beliefs converge to the desired (and maximum a posteriori) estimations. Historically, the BP method was developed for tree graphs (A graph that any two nodes are connected by exactly one path. In factor graph demonstrations, it means that the factor graphs have no in-and-itself cycles [55]). Their message passing was intended such that the beliefs would converge after one shot of message passings from a root to leaf nodes [56]. Later, in more general cases and more complex constructions, for example, cases that include loops, the updating rule is not the simple routine anymore and needs some considerations and correctness to can implement the BP in arbitrary (loop included) cases [57] [58] [59] [60]. One of the early research projects that consider the other graph forms, rather than tree graphs, and surveys their convergence is [57], which raises the question of "does loopy propagation work in more general settings?" Four Bayesian network architectures are being analysed, including the architectures of ALARM and QMR networks (respectively, Figure 3.7 and Figure 3.8). Often loopy belief propagation gives a reasonable estimation of correct marginals. However, there is some oscillation at loopy message passing on the QMR network, and this paper investigates the cause of these oscillations. Besides that, it also proposes some methods to prevent these wrong results.

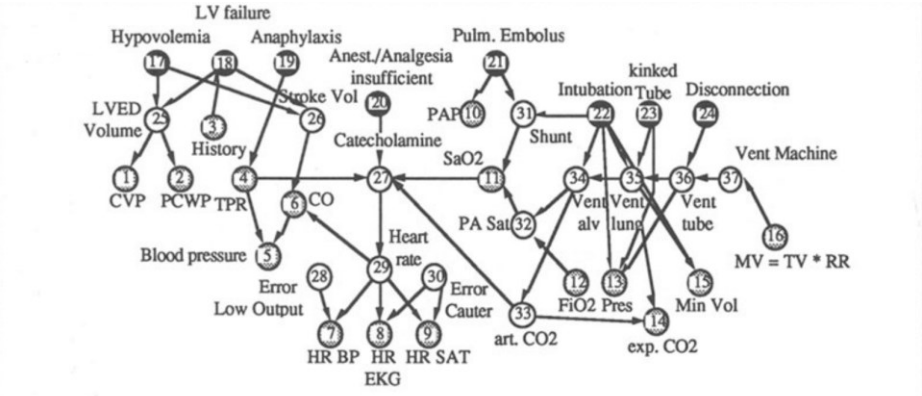


Figure 3.7 An example of ALARM architectures that had been used as an alarm message system for patient monitoring [57].

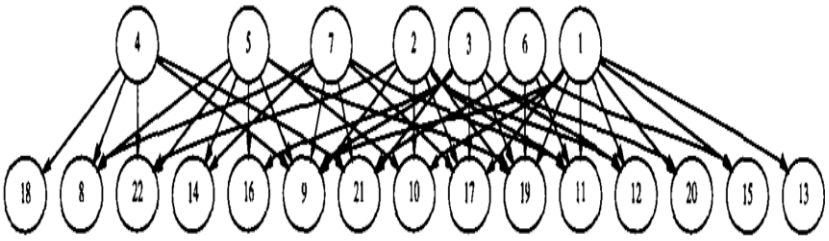


Figure 3.8 An example network that demonstrates The QMR structure, which generally is a bipartite structure [57].

3.1.3. Belief Propagation as the Variational Inference Method Approximation

In more advanced studies about factor graphs, such as [60], loopy belief propagation is treated in different approaches. Instead of making the MAP inference (that is related to minimizing all the factor energies), the inference would stay as an optimization problem. Loopy belief propagation, as an approximate variational inference method, tries to minimize the Kullback–Leibler divergence between the approximated variational distribution and the posterior probability distributions.

This Kullback–Leibler divergence approximation, in the case of graphs MAP interference approximations, is recognized as the minimization over the Bethe free energy [61], which generally is not a convex function. This means that loopy belief propagation cannot guarantee that the beliefs converge to MAP inference. Even in converging iterations, this convergence might be to local optimum marginals, which are not the same as the global and actual ones. However, hopefully, since we are interested in modeling the available states and observations with Gaussian distributions, and since there is a guarantee that loopy belief propagation for the Gaussian systems would converge to exact marginals in converging cases, they would not be any miss-converging to wrong marginals by utilizing Gaussian belief propagations.

However, this property should not be translated as a perpetual converging in all arbitrary graphs and message passing methods. Gaussian systems would be confident that their mapping on graphs and applying Gaussian belief propagation would not lead to convergence to MAP inference. Thus, some researchers [62] [63] [64] study the conditions and methods that can increase convergence possibilities in GBP cases.

3.2. Gaussian Belief Propagation

After discussing the convergence for factor graphs with all Gaussian states, we will look at these Gaussian factor graph systems in more detail in this session. When we talk about the Gaussian model in a factor graph, all factors and, consequently, the joint

posterior have Gaussian distribution (with single or multiple random variables). There could be some benefits that come with accepting Gaussian distribution:

1. This model suits well to describe actual observations and could be a proper representative of real word states.
2. When it comes to calculating the marginalization, conditioning, or products, this distribution is inexpensive in terms of computation complexity.

As we mentioned earlier, one of the popular ways to interpret the factor nodes is to define an energy function while $f(X_j) \propto \exp\left(-E(X_j)\right)$. The belief propagation also could be explained as algorithms that end up reaching marginal X_{MAP} by minimizing the sum of all available energy functions. In other words:

$$X_{MAP} = \arg min_X \left(\sum_i E_i(X_i) \right) \qquad \text{Equation 3.12}$$

Now, in Gaussian space, as we know, the general way to describe the random variables (in both univariate and multivariate) is: $p(X) \propto \exp\left(-\frac{1}{2}(x-\mu)^\intercal \Sigma^{-1}(x-\mu)\right)$, and according to the energy function interpretation, we can clarify $\frac{1}{2}(x-\mu)^\intercal \Sigma^{-1}(x-\mu)$ as the quadratic energy function that must be minimized.

Since for Gaussian distribution, there are two distinct forms for describing the exponential function (the Moments or the Canonical forms) there are two ways to employ Gaussian distributions in different situations (due to each form's properties) [55]. Usually, in the marginalization operations, as we would see, it is easier to work with the Moments form. On the other hand, it is easy to use the Canonical form for conditioning and taking PDF products.

By expanding the Moments form, we can find the main necessary relations to relate the Moment and Canonical forms:

$$\begin{aligned} E(x) &= \frac{1}{2}[(x-\mu)^\intercal \Sigma^{-1}(x-\mu)] = \frac{1}{2}[(x^\intercal \Sigma^{-1} - \mu^\intercal \Sigma^{-1}) \times (x-\mu)] \\ &= \frac{1}{2}[(x^\intercal \Sigma^{-1} x - \mu^\intercal \Sigma^{-1} x) - (x^\intercal \Sigma^{-1}\mu - \mu^\intercal \Sigma^{-1}\mu)] \end{aligned} \qquad \text{Equation 3.13}$$

since $\mu^\intercal \Sigma^{-1} x = x^\intercal \Sigma^{-1}\mu =$ a scalar value:

$$\begin{aligned} \rightarrow E(x) &= \frac{1}{2}\left[\left(x^\intercal \Sigma^{-1} x - 2\mu^\intercal \Sigma^{-1} x\right) + \mu^\intercal \Sigma^{-1}\mu\right] \\ &= \left(\frac{1}{2} x^\intercal \Sigma^{-1} x - \mu^\intercal \Sigma^{-1} x\right) + \frac{1}{2}\left(\mu^\intercal \Sigma^{-1}\mu\right) \end{aligned} \qquad \text{Equation 3.14}$$

it is possible to ignore the term $\frac{1}{2}(\mu^\top\Sigma^{-1}\mu)$ because it is independent of the random variables X.

Since:

$$\left(\frac{1}{2}x^\top\Sigma^{-1}x - (\Sigma^{-1}\mu)^\top x\right) = \left(\frac{1}{2}x^\top\Lambda x - \eta^\top x\right) \quad \text{Equation 3.15}$$

And it leads to:

$$\Sigma^{-1} = \Lambda \quad \text{Equation 3.16 Covariance and Precision Matrix Relation}$$

$$\eta = \Sigma^{-1}\mu \quad \text{Equation 3.17 Mean and Information vectors Relation}$$

Based on the Hammersley-Clifford theorem, we know that for all existing variates set in a factor graph, which make the main clique, the primary joint distribution $p(X)$ can be factorized as:

$$p(X) = \prod_i f_i(X_i) \quad \text{Equation 3.18}$$

Because, based on complete Gaussian modeling, all the factor nodes $f_j(X_j)$ have Gaussian distributions, the joint distribution corresponding to a factor graph can also be presented as a Gaussian distribution, that in the Canonical form it would be:

$$p(X) \propto \exp\left(-\frac{1}{2}X^\top\Lambda X + \eta^\top X\right) \quad \text{Equation 3.19}$$

And its corresponding energy function also would be:

$$\mathrm{E(X)} = \frac{1}{2}X^\top\Lambda X - \eta^\top X \quad \text{Equation 3.20}$$

Since MAP inference would minimize the energy of the whole system, we can trivially find MAP states by minimizing the multivariate quadratic energy function [55]:

$$\nabla_X E = \nabla_X \log P(X) = -\Lambda X + \eta \quad \text{Equation 3.21.a}$$

$$for\ \nabla_X E = 0 \longrightarrow -\Lambda X + \eta = 0 \longrightarrow \Lambda X = \eta \quad \text{Equation 3.21.b}$$

$$\longrightarrow X^{MAP} = \Lambda^{-1}\eta \xrightarrow{\eta=\Lambda\mu} X^{MAP} = \mu \quad \text{Equation 3.21.c}$$

This realization (Equation 3.21.a.c) can be defined this way after completing the Gaussian belief propagation (and expectantly by converging to optimum marginals) this optimum would illustrate itself as the mean vector of a multivariate distribution.

3.2.1. Marginal Gaussian Distribution Properties

After obtaining the $p(\boldsymbol{X})$, it is time to apply the marginalization to obtain the information matrix and precision matrix (or the mean vector and the covariance matrix) for each variable node. Based on the definition, the marginal distribution is given by:

$$p(\boldsymbol{X}_i) = \int p(\boldsymbol{X}) d\boldsymbol{X}_{-i} \qquad \text{Equation 3.22}$$

And by knowing the mean vector and the covariance matrix, the marginal PDF can easily be obtained as the: $p(\mathbf{X}_i) = \mathcal{N}(\mathbf{X}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{ii})$. However, the challenge in marginalization is that in Gaussian belief updating (that we will discuss later), taking products would be done in the Canonical forms, and we must extract the marginal covariance matrix from the precision matrix. Obtaining these matrixes should not be too hard for variable nodes. Because for the equation we obtained in Equation 3.16 ($\Sigma^{-1} = \Lambda \Leftrightarrow \Lambda^{-1} = \Sigma$) After doing the belief updating and obtaining renewed precision matrixes, the covariance matrix can be found clearly and subdivided in the following way to obtain the wanted vector and matrix:

For covariance vector:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1} \xrightarrow{\textit{doing the required operations}} \boldsymbol{\Sigma} \xrightarrow{\textit{Partitioning}} \begin{pmatrix} \boldsymbol{\Sigma}_{i\times i} & \boldsymbol{\Sigma}_{i\times -i} \\ \boldsymbol{\Sigma}_{-i\times i} & \boldsymbol{\Sigma}_{-i\times -i} \end{pmatrix} \qquad \text{Equation 3.23}$$

Similarly, for the mean vector:

$$\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} \xrightarrow{\textit{doing inverse and multiplication operations on } \Lambda \,\&\, \eta} \boldsymbol{\mu} \xrightarrow{\textit{Partitioning}} \begin{pmatrix} \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_{-i} \end{pmatrix} \qquad \text{Equation 3.24}$$

Then, we can simply express marginalized $\boldsymbol{X}_i$variates $p(\mathbf{X}_i) = \mathcal{N}(\mathbf{X}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{ii})$

3.2.2. Non-Gaussian Factors (Non-Quadratic Energy Functions)

So far, we have assumed that all the observations, states, and factors follow the Gaussian distribution and that it is possible to implement the Gaussian belief propagation algorithm. Now, it is time to involve more realistic relations between variables nodes. The relations that necessarily are not Gaussian-based models. Therefore, in this section, we are trying to present some rearrangement or linearization approaches that would make it feasible to fit more energy functions in the Gaussian

suitable quadratic form. First, let us look at the linear energy functions and see how they can be presented in the Moments or the Canonical forms:

Assume that we attempt to optimize the variable nodes' states: $X_i, X_{i+1}, \dots, X_j$ in a way to minimize the energy function of kth factor node $h_k(\cdot)$:

$$\arg min_{X_i, X_{i+1}, \dots, X_j} \quad \left| h_k(X_i, X_{i+1}, \dots, X_j) \right| \qquad \text{Equation 3.25}$$

To present this function in some preferred forms, we should assign a covariance or precision matrix to the output of energy functions. This matrix can refer to the correlation between different output parameters of $h_k(\cdot)$, or the precision of these outputs. In the cases that $h_k(\cdot)$ describes the observation or measurement activities, this covariance matrix relates to the noises and accuracy of measurements. This matrix should be defined based on all incorporating nodes' variables $\{X_i, X_{i+1}, \dots, X_j\}$. After assigning a covariance matrix $\mathbf{\Sigma}$, it would be doable to describe the factor functions as a Mahalanobis distance:

$$f_k(\mathbf{X}) \propto e^{\left(-\frac{1}{2}\left\| h_k(X_i, X_{i+1}, \dots, X_j) \right\|_{\mathbf{\Sigma}}\right)} = e^{\left(-\frac{1}{2} h_k(X_i, X_{i+1}, \dots, X_j)^{\intercal} \mathbf{\Sigma^{-1}} h_k(X_i, X_{i+1}, \dots, X_j)\right)} \qquad \text{Equation 3.26}$$

Now, it is time to reform the factor functions. If we assume that the energy function has a linear form, $h_k(\boldsymbol{X}) = H \cdot \boldsymbol{X} + b$:

$$\begin{aligned} h_k(\boldsymbol{X})^{\intercal} \cdot \mathbf{\Sigma^{-1}} \cdot h_k(\mathbf{X}) &= (H \cdot \boldsymbol{X} + b)^{\intercal} \mathbf{\Sigma^{-1}} (H \cdot \boldsymbol{X} + b) \\ &= (\boldsymbol{X}^{\intercal} \cdot H^{\intercal} + b^{\intercal}) \mathbf{\Sigma^{-1}} (H \cdot \boldsymbol{X} + b) \\ &= \boldsymbol{X}^{\intercal} \cdot H^{\intercal} \mathbf{\Sigma^{-1}} H \cdot \boldsymbol{X} - 2(-b^{\intercal} \mathbf{\Sigma^{-1}} H) \cdot \boldsymbol{X} + b^{\intercal} \mathbf{\Sigma^{-1}} b \end{aligned} \qquad \text{Equation 3.27}$$

By disregarding constant term $b^{\intercal} \mathbf{\Sigma^{-1}} b$, the linear energy functions can be reformed in the Canonical form as:

$$E(\boldsymbol{X}) = \frac{1}{2} \boldsymbol{X}^{\intercal} \Lambda' \boldsymbol{X} - \eta'^{\intercal} \boldsymbol{X} \qquad \text{Equation 3.28.a}$$

Where:

$$\Lambda' = H^{\intercal} \mathbf{\Sigma^{-1}} H \qquad \text{Equation 3.28.b}$$

And:

$$\eta' = -\mathrm{H}^{\intercal} \mathbf{\Sigma^{-1}} b \qquad \text{Equation 3.28.c}$$

Therefore, this function can be employed in the Gaussian belief propagation procedure.

The other cases that must be evaluated are nonlinear $h_k(X)$. For these cases, as [65] suggests, obtaining the first order Taylor expansion around the current variable nodes'

states estimations $\{X_i, X_{i+1}, \dots, X_j\}$, and using the acquired linearize function could be a suitable approach to approximate the factor as the desired Gaussian.

Based on Taylor series expression, for the current state $\boldsymbol{X}^0 = \{\boldsymbol{X}_i^0, \boldsymbol{X}_{i+1}^0, \dots, \boldsymbol{X}_j^0\}$ and the obtained Jacobian matrix J, which its partitioned structure would be (for assuming factor function has a n dimensional output $\mathbf{f} = [\boldsymbol{f_1} \quad \boldsymbol{f_2} \quad \cdots \quad \boldsymbol{f_n}]$):

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \boldsymbol{X}_i} & \cdots & \frac{\partial \mathbf{f}}{\partial \boldsymbol{X}_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{X}_i} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{X}_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \boldsymbol{X}_i} & \cdots & \frac{\partial f_n}{\partial \boldsymbol{X}_j} \end{bmatrix} \qquad \text{Equation 3.29}$$

The Taylor expansion is:

$$h_k(\boldsymbol{X}) \approx h_k(\boldsymbol{X_0}) + \mathbf{J} \cdot (\boldsymbol{X} - \boldsymbol{X_0}) \xrightarrow{\Delta \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X_0}} h_k(\boldsymbol{X}) \approx \mathbf{J} \cdot (\Delta \boldsymbol{X}) + h_k(\boldsymbol{X_0}) \qquad \text{Equation 3.30}$$

By equating the achieved Taylor series with the analysed linear form $H \cdot \boldsymbol{X} + b$, we can determine the information vector and precision matrix:

$$\begin{cases} H = \mathbf{J} \\ b = h_k(\boldsymbol{X_0}) \end{cases} \rightarrow \begin{cases} \Lambda' = \mathbf{J}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{J} \\ \eta' = -\mathbf{J}^\intercal \boldsymbol{\Sigma}^{-1} h_k(\boldsymbol{X_0}) \\ \Delta \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X_0} \end{cases} \qquad \text{Equation 3.31}$$

$$\rightarrow h_k(\boldsymbol{X}) \approx \Delta \boldsymbol{X}^\intercal \cdot \mathbf{J}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{J} \cdot \Delta \boldsymbol{X} - 2\left(-h_k(\boldsymbol{X_0})^\intercal \boldsymbol{\Sigma}^{-1} H\right) \cdot \Delta \boldsymbol{X}$$

By using the found technique, it would be possible to linearize a general nonlinear factor $h_k(\boldsymbol{X})$ around specific variables states' $\boldsymbol{X}$, for turning the function into a Gaussian expression.

3.2.3. Gaussian Belief Propagation Equations

As we stated earlier, there are three phases in applying belief propagation. Now, we try to modify these steps along with the Gaussian states modeling. As we would see, the advantage of utilizing the Gaussian model is that all the multiplications will be converted to summations that not only make the belief propagation lighter in terms of computation resources, but the equations will also be converted into more straightforward phases. Coming next, we will compute and conclude these phases for Gaussian models.

- **Factor-to-Variable Gaussian Message Passing:**
  As explained in prior sessions, this phase includes message gathering and marginalization, and the final formula would be:

$$m_{f_o \to x_i} = \sum_{X_o \setminus x_i} \left(f_j(X_j) \times \prod_{k \in N(o) \setminus i} m_{x_k \to f_o} \right) \qquad \text{Equation 3.32}$$

As we sorted variables of variable nodes are connected to the factor graph f_o as $X_o = \{X_i, X_{i+1}, \dots, X_j\}$, for coming messages from each corresponding variable nodes $(i, i+1, \dots, j)$, the messages only are the functions of variables corresponding to source nodes. In other words, $m_{x_k \to f_o}(X) = m_{x_k \to f_o}(X_k)$. Therefore, if we display the precision matrix and information vectors presenting the current state of factor nodes as the following partitioned ones:

$$\boldsymbol{\Lambda}_{f_o} = \begin{bmatrix} \boldsymbol{\Lambda}_{f_{i,i}} & \dots & \boldsymbol{\Lambda}_{f_{i,k}} & \dots & \boldsymbol{\Lambda}_{f_{i,j}} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \boldsymbol{\Lambda}_{f_{k,i}} & \dots & \boldsymbol{\Lambda}_{f_{k,k}} & \dots & \boldsymbol{\Lambda}_{f_{k,j}} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{f_{j,i}} & \cdots & \boldsymbol{\Lambda}_{f_{j,k}} & \dots & \boldsymbol{\Lambda}_{f_{j,j}} \end{bmatrix}, \boldsymbol{\eta}_{f_o} = \begin{bmatrix} \boldsymbol{\eta}_{f_i} \\ \vdots \\ \boldsymbol{\eta}_{f_k} \\ \vdots \\ \boldsymbol{\eta}_{f_j} \end{bmatrix} \qquad \text{Equation 3.33}$$

After executing the multiplication step, only for k_{th} variable node $\left(f_j(X_j) \times m_{x_k \to f_o}\right)$, the vector and matrix at this step are adjusted to:

$$\boldsymbol{\Lambda}_{f_{x_k \to f_o}} = \begin{bmatrix} \boldsymbol{\Lambda}_{f_{i,i}} & \dots & \boldsymbol{\Lambda}_{f_{i,k}} & \dots & \boldsymbol{\Lambda}_{f_{i,j}} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \boldsymbol{\Lambda}_{f_{k,i}} & \dots & \boldsymbol{\Lambda}_{f_{k,k}} + \boldsymbol{\Lambda}_{x_k \to f_o} & \dots & \boldsymbol{\Lambda}_{f_{k,j}} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{f_{j,i}} & \cdots & \boldsymbol{\Lambda}_{f_{j,k}} & \dots & \boldsymbol{\Lambda}_{f_{j,j}} \end{bmatrix}$$

$$\boldsymbol{\eta}_{f_{x_k \to f_o}} = \begin{bmatrix} \boldsymbol{\eta}_{f_i} \\ \vdots \\ \boldsymbol{\eta}_{f_k} + \boldsymbol{\eta}_{x_k \to f_o} \\ \vdots \\ \boldsymbol{\eta}_{f_j} \end{bmatrix} \qquad \text{Equation 3.34}$$

Subsequently, after applying the received messages from all variable nodes (but i_{th}) the ultimate shape for precision matrix and information vector would be shifted to:

$$\boldsymbol{\Lambda}_{f_{All}} = \begin{bmatrix} \boldsymbol{\Lambda}_{f_{i,i}} & \dots & \boldsymbol{\Lambda}_{f_{i,k}} & \dots & \boldsymbol{\Lambda}_{f_{i,j}} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \boldsymbol{\Lambda}_{f_{k,i}} & \dots & \boldsymbol{\Lambda}_{f_{k,k}} + \boldsymbol{\Lambda}_{x_k \to f_o} & \dots & \boldsymbol{\Lambda}_{f_{k,j}} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{f_{j,i}} & \cdots & \boldsymbol{\Lambda}_{f_{j,k}} & \dots & \boldsymbol{\Lambda}_{f_{j,j} + \boldsymbol{\Lambda}_{x_j \to f_o}} \end{bmatrix} \qquad \text{Equation 3.35}$$

$$\boldsymbol{\eta}_{f_{All}} = \begin{bmatrix} \boldsymbol{\eta}_{f_i} \\ \vdots \\ \boldsymbol{\eta}_{f_k} + \boldsymbol{\eta}_{x_k \to f_j} \\ \vdots \\ \boldsymbol{\eta}_{f_j} + \boldsymbol{\eta}_{x_j \to f_o} \end{bmatrix} \qquad \text{Equation 3.36}$$

Next, it is time for marginalization over all variables but ith. For this matter, we would utilize the following identity for obtaining the inverse of a partitioned matrix [66] (Chapter 2.3.1) to compute $\boldsymbol{\Sigma}_{i\times i}$ by using achieved precision matrix $\boldsymbol{\Lambda}_{f_{All}}$ at Equation 3.35:

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{M} & -\boldsymbol{MBD}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{CM} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{CMBD}^{-1} \end{pmatrix} \qquad \text{Equation 3.37}$$
$$\boldsymbol{while}\ \boldsymbol{M} = (\boldsymbol{A} - \boldsymbol{BD}^{-1}\boldsymbol{C})^{-1}$$

With using Equation 3.16 and Equation 3.37, and by partitioning the available variables as $\boldsymbol{X}_i$ & $\boldsymbol{X}_{-i}$ we have the following precision matrix $\boldsymbol{\Lambda}_{f_{All}}$:

$$\boldsymbol{\Lambda}_{f_{All}} = \begin{pmatrix} \boldsymbol{\Lambda}_{i\times i} & \boldsymbol{\Lambda}_{i\times -i} \\ \boldsymbol{\Lambda}_{-i\times i} & \boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}} \end{pmatrix} \qquad \text{Equation 3.38}$$

And since $\Sigma = \Lambda^{-1}$:

$$\begin{pmatrix} \boldsymbol{\Sigma}_{i\times i} & \boldsymbol{\Sigma}_{i\times -i} \\ \boldsymbol{\Sigma}_{-i\times i} & \boldsymbol{\Sigma}_{-i\times -i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Lambda}_{i\times i} & \boldsymbol{\Lambda}_{i\times -i} \\ \boldsymbol{\Lambda}_{-i\times i} & \boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}} \end{pmatrix}^{-1} \qquad \text{Equation 3.39}$$

By applying the stated property about the inverse of the partitioned matrix:

$$\begin{aligned} &\boldsymbol{\Sigma}_{f_o \to m_{X_{-i}}} = \left(\boldsymbol{\Lambda}_{i\times i} - \boldsymbol{\Lambda}_{i\times -i}\left(\boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}}\right)^{-1}\boldsymbol{\Lambda}_{-i\times i}\right)^{-1} \\ &\to \boldsymbol{\Lambda}_{f_o \to m_{X_{-i}}} = \boldsymbol{\Lambda}_{i\times i} - \boldsymbol{\Lambda}_{i\times -i}\left(\boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}}\right)^{-1}\boldsymbol{\Lambda}_{-i\times i} \\ &\xrightarrow{\Sigma=\Lambda^{-1}} \boldsymbol{\Sigma}_{f_o \to m_{X_{-i}}} = \boldsymbol{\Sigma}_{i\times i} - \boldsymbol{\Sigma}_{-i\times i}\left(\boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}}\right)\boldsymbol{\Sigma}_{i\times -i} \end{aligned} \qquad \text{Equation 3.40}$$

Similarly, for the information vector $\boldsymbol{\eta}$:

$$\boldsymbol{\eta}_{f_o \to m_{X_{-i}}} = \boldsymbol{\eta}_{i\times i} - \boldsymbol{\Lambda}_{i\times -i}\left(\boldsymbol{\Lambda}_{-i\times -i} + \boldsymbol{\Lambda}_{m_{X_{-i}\to f_o}}\right)^{-1}\boldsymbol{\eta}_{-i\times i}$$

- **Variable-to-factor Gaussian message passing:**
  As we explored general Variable-to-factor messaging, the generated message in variable node ith to pass it to factor node oth is being made by:

$$m_{x_i \to f_o} = \prod_{s \in N(i)\setminus o} m_{f_s \to x_i} \qquad \text{Equation 3.41}$$

Since we have already switched to Gaussian models, we know that all the messages have Gaussian forms $m_{f_s \to x_i} \propto \mathcal{N}^{-1}(\eta_{f_s \to x_i}, \Lambda_{f_s \to x_i})$. So, the message aggregations would be converted to summations on the precision matrixes and the information vectors. It could simply be confirmed in this way:

$$\begin{aligned} m_{x_i \to f_o} &= \prod_{s \in N(i) \setminus o} m_{f_s \to x_i} \propto \prod_{s \in N(i) \setminus o} \mathcal{N}^{-1}(\boldsymbol{\eta}_{f_s \to x_i}, \boldsymbol{\Lambda}_{f_s \to x_i}) \\ &= \prod_{s \in N(i) \setminus o} \left(\frac{1}{2} \boldsymbol{X}_i^{\intercal} \boldsymbol{\Lambda}_{f_s \to x_i} \boldsymbol{X}_i - \boldsymbol{\eta}_{f_s \to x_i}^{\intercal} \boldsymbol{X}_i \right) \\ &= \frac{1}{2} \boldsymbol{X}_i^{\intercal} \sum_{s \in N(i) \setminus o} \boldsymbol{\Lambda}_{f_s \to x_i} \boldsymbol{X}_i - \left(\sum_{s \in N(i) \setminus o} \boldsymbol{\eta}_{f_s \to x_i} \right)^{\intercal} \boldsymbol{X}_i \end{aligned} \quad \text{Equation 3.42}$$

Therefore:

$$\boldsymbol{\Lambda}_{m_{x_i \to f_o}} = \sum_{s \in N(i) \setminus o} \boldsymbol{\Lambda}_{f_s \to x_i} \quad \text{Equation 3.43}$$

$$\boldsymbol{\eta}_{m_{x_i \to f_o}} = \sum_{s \in N(i) \setminus o} \boldsymbol{\eta}_{f_s \to x_i} \quad \text{Equation 3.44}$$

- **Gaussian belief updating:**
  Renewing variable node state is being done by gathering all the received messages in this way:

$$b_i(\boldsymbol{X}_i) = \prod_{s \in N(i)} m_{f_s \to x_i} \quad \text{Equation 3.45}$$

As same as the previous section, by rewriting the belief propagation equation in the Canonical forms, the updated belief would be:

$$\begin{aligned} b_i(\boldsymbol{X}_i) &= \prod_{s \in N(i)} m_{f_s \to x_i} = \prod_{s \in N(i)} \left(\frac{1}{2} \boldsymbol{X}_i^{\intercal} \boldsymbol{\Lambda}_{f_s \to x_i} \boldsymbol{X}_i - \boldsymbol{\eta}_{f_s \to x_i}^{\intercal} \boldsymbol{X}_i \right) \\ &= \frac{1}{2} \boldsymbol{X}_i^{\intercal} \sum_{s \in N(i)} \boldsymbol{\Lambda}_{f_s \to x_i} \boldsymbol{X}_i - \left(\sum_{s \in N(i)} \boldsymbol{\eta}_{f_s \to x_i} \right)^{\intercal} \boldsymbol{X}_i \end{aligned} \quad \text{Equation 3.46}$$

Therefore, the belief parameters $\boldsymbol{\eta}_{b_i}$ and $\boldsymbol{\Lambda}_{b_i}$ are:

$$\boldsymbol{\eta}_{b_i} = \sum_{s \in N(i)} \boldsymbol{\eta}_{f_s \to x_i} \; and \; \boldsymbol{\Lambda}_{b_i} = \sum_{s \in N(i)} \boldsymbol{\Lambda}_{f_s \to x_i} \quad \text{Equation 3.47}$$

3.3. Vehicles Motion Plannings as an Imperfect Collaborative Network

As our earlier discussions, multiple AI-empowered and autonomous vehicles, in different traffic scenes with the chance of numerous different actions and behaviors, will need to cooperate in terms of sharing their own perceived insight and observation for the sake of achieving safety and efficiency in driving and lowering the chance of any collisions that can happen for the lack of vital perception and awareness about their surrounding environment. We discussed that establishing this issue as the main goal for organizing our communications and orienting the required strategies could have good potential for performing suitable transmission policies that attempt to reduce the collision rates subject to probable network constraints. Moreover, proposing well-performed strategies will have enough potential to prevent frequent congestions that can be occurred due to over-transmitting redundant, raw, or ineffective processed data.

After reviewing the sophisticated train graph neural network, the SpAGNN, which has an astonishing ability to predict vehicle behaviors in different road networks, we have analyzed the Gaussian belief propagation method, which was the primary motivation for designing this neural network. After investigating this technique, we find out that message passing iterations in these networks are to make an inference that minimizes the comprehensive energy functions of factor nodes that identify the type of interactions between variable nodes. More importantly, thanks to applying the quadratic energy function (which leads to factor nodes with Gaussian forms), we know that by taking the right message passing plan that guarantees the convergence of the states, the final converged states would be the optimal ones that would return the most minimalized energy functions outputs.

In this session, we will propose a proper dynamic factor graph network that demonstrates the estimated interactions of existing objects in a traffic scene. Assessing this factor graph network would help us predict these objects' short-term behaviors. The main idea of this graph modeling (and prediction) is to minimize the basic linear interaction functions between interacted vehicles. This basic modeling takes (even minimal) account of each vehicle's coordination to estimate their future decision in a collision-avoiding manner. Meaning that, for robust coordination, it is possible to expect precise behavior forecasting to avoid collisions. Conversely, these graphs anticipated collisions will nearly take place without proper cooperation (or awareness) between vehicles, so they are also trustworthy and reliable predictions. The causes of these predicted collisions must be targeted and solved beforehand so as to avoid all possible collisions.

One of the research studies characterizing the dynamic interactions between several objects by graph factor networks is [67] , which formulates multi-robot planning as a dynamic optimization problem. The defined constraint factors that demonstrate the type of relations among different objects could be so helpful in our studies as well, and utilizing them for characterizing the cooperative perception could be so beneficial.

The main goal of [67] is to formulate the multi-robot trajectory planning based on least squares minimization of some defined energy functions, which is equivalent to presenting inference on a Gaussian factor graph and applying Gaussian belief propagation to reach optimum states. Therefore, they proposed a factor graph with defined variables and factor nodes to model this multi-robot planning and the possible multi-robot interactions in their studies. The following diagram (Figure 3.9) illustrates the [67] proposed characterization for an example of two robots planning. Later, we are going to investigate their proposed factor nodes (with respect to the presented form of states nodes)

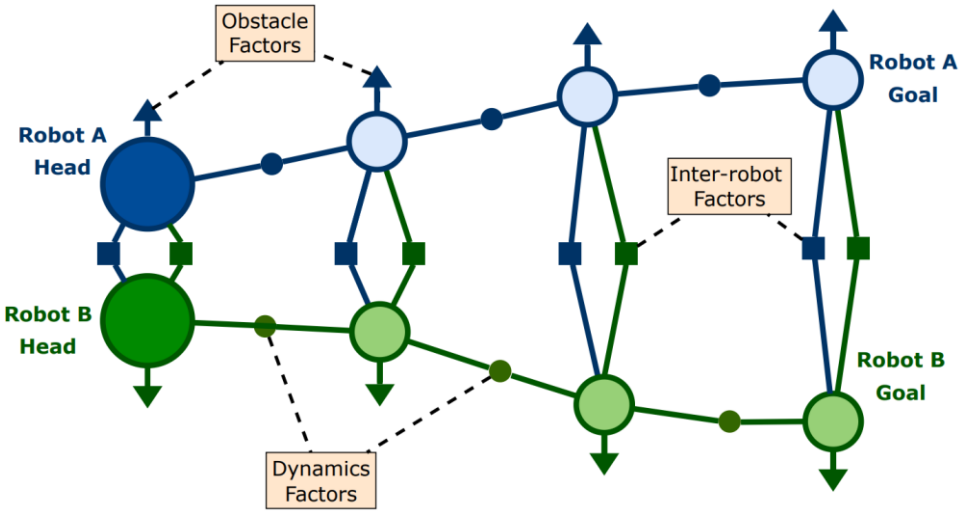


Figure 3.9 An example of proposed factor graph characterization in [67]. In this example, there are two robots whose corresponding various types of factors are presented.

The presented scenario in this study is a 2-D scene that all robots have two degrees of movement freedom. This scenario is like the BEV map used in earlier mentioned networks, such as the SpAGNN. As it is noticeable in the following figured diagram, each robot is represented with different variable nodes for different timesteps. This means all variable nodes are time-dependent and state the position and velocity of vehicles at the corresponding timestep t_k:

$$\mathbf{x}_k = [\boldsymbol{x}_k^\top, \dot{\boldsymbol{x}}_k^\top]^\top$$

$$\begin{cases} \boldsymbol{x}_k \; as \; position \; states: \begin{bmatrix} x_k \\ y_k \end{bmatrix} \\ \dot{\boldsymbol{x}}_k \; as \; velocity \; states: \begin{bmatrix} \dot{x}_k \\ \dot{x}_k \end{bmatrix} \end{cases} \rightarrow Variable \; Motion \; State \; \mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} \qquad \text{Equation 3.48}$$

- **Proposed Factor Graphs:**
  There are four innovative factor graphs that propose by [67] to have the most accurate and practical short-term representation of robots moving and their interactive planning. These four factors are:
  a. Pose factor
  b. Dynamic factor
  c. Obstacle factor
  d. Inter-robot factor

a. Pose Factor f_p:

Study [67] presumes that the initial (current) state and the finishing state (that is called as horizon state) are measured and expected in this planning method (how does each robot start, and how is it supposed to finish its maneuver). These identifying states are employed in the planning strategy for each specific robot by linking two pose factors to each robot's first and last variable nodes. Implanting these factors is for "anchoring" these important states at optimization. Pose factor has a Gaussian innate form that can be described (in the Moments form) as:

$$Mean\ Vector\ \boldsymbol{\mu} = \boldsymbol{X}_k \quad \text{Equation 3.49}$$

$$Observation\ Covariance\ \boldsymbol{\Sigma} = \sigma_p^2 \times \boldsymbol{I}_{4\times4} \quad \text{Equation 3.50}$$

Furthermore, this model can be displayed in the Canonical form:

$$Observation\ Precision\ \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \sigma_p^{-2} \times \boldsymbol{I}_{4\times4} \quad \text{Equation 3.51}$$

$$Information\ Vector\ \boldsymbol{\eta} = \boldsymbol{\Lambda\mu} = \sigma_p^{-2} \times \boldsymbol{X}_k \quad \text{Equation 3.52}$$

Intuitively, desiring more robust anchoring leads to more precise observations of initial information, with a tinier standard deviation σ_p that makes up more rigid and insensitive trajectories.

b. Dynamic Factors f_d:

As Figure 3.9 exhibits, successive variable nodes of a robot that express its dynamic states for sequential timesteps are connected by dynamic factors. These factors guarantee the states' smooth sweeping over time. [67] composes its factor energy function linearly (for two inputs corresponding to two connected successive variable nodes)

$$\boldsymbol{h}_d(\mathbf{X}_k, \mathbf{X}_{k+1}) = \boldsymbol{\Phi}(t_{k+1}, t_k)\mathbf{X}_k - \mathbf{X}_{k+1} \quad \text{Equation 3.53}$$

Where $\mathbf{\Phi}$ is the transition matrix from time t_k to t_{k+1} ($\Delta t_k = t_{k+1} - t_k$):

$$\mathbf{\Phi}(\Delta t_k) = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Equation 3.54}$$

The [67] also defines the covariance matrix for dynamic factors obtained from a "noise-on-acceleration" model. For defined acceleration noise (in both x and y axes) $\boldsymbol{Q}_d = \sigma_d^2 \boldsymbol{I}_{\mathbf{2}\times\mathbf{2}}$:

$$\mathbf{\Sigma}_d(\Delta t_k) = \begin{bmatrix} \frac{1}{3}\Delta t_k^3 \mathbf{Q}_d & \frac{1}{2}\Delta t_k^2 \mathbf{Q}_d \\ \frac{1}{2}\Delta t_k^2 \mathbf{Q}_d & \Delta t_k \mathbf{Q}_d \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\Delta t_k^3\sigma_d^2 & 0 & \frac{1}{2}\Delta t_k^2\sigma_d^2 & 0 \\ 0 & \frac{1}{3}\Delta t_k^3\sigma_d^2 & 0 & \frac{1}{2}\Delta t_k^2\sigma_d^2 \\ \frac{1}{2}\Delta t_k^2\sigma_d^2 & 0 & \Delta t_k\sigma_d^2 & 0 \\ 0 & \frac{1}{2}\Delta t_k^2\sigma_d^2 & 0 & \Delta t_k\sigma_d^2 \end{bmatrix} \quad \text{Equation 3.55}$$

For dynamic factor $\boldsymbol{h}_d(\mathbf{X}_k, \mathbf{X}_{k+1}) = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{X}_k - \mathbf{X}_{k+1}$, the equal $h(\boldsymbol{X}) = H \cdot \boldsymbol{X} + \boldsymbol{b}$ representation would be:

$$H = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t_k & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_k \\ \boldsymbol{X}_{k+1} \end{bmatrix}, \boldsymbol{b} = \emptyset \quad \text{Equation 3.56}$$

Then, for $\mathbf{\Lambda}_d = \mathbf{\Sigma}_{\boldsymbol{d}}^{-\mathbf{1}}(\Delta t_k)$ as well as Equation 3.28, the equivalent $\mathbf{\Lambda}'$ and $\boldsymbol{\eta}'$would be:

$$\boldsymbol{\Lambda}'_{\mathbf{8}\times\mathbf{8}} = H^\top \boldsymbol{\Lambda}_d H \quad \text{Equation 3.57}$$

$$\boldsymbol{\eta}'_{\mathbf{8}\times\mathbf{1}} = -H^\top \boldsymbol{\Lambda}_d \emptyset_{\mathbf{4}\times\mathbf{1}} = \emptyset_{\mathbf{8}\times\mathbf{1}} \quad \text{Equation 3.58}$$

c. Inter-Robot Factors f_r:

The Inter-Robot factor is the main factor graph that simulates the real vehicle collision avoiding interaction. At each timestep, the factor node relates two variable nodes of two different robots bilaterally. The input of energy function of these factor nodes is the distance between two interacted robots (that would be calculated by their current states $\boldsymbol{X}_k^A$, $\boldsymbol{X}_k^B$). The energy of Inter-Robot factor nodes will be non-zero if the robots' distance is less than the critical defined distance r^*. [67] prefers to define the energy function as the following equation, generating zero outputs for distances more than the critical distance r^*:

$$\boldsymbol{h}_r(\boldsymbol{x}_k^A, \boldsymbol{x}_k^B) = \begin{cases} 1 - \dfrac{distance(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B)}{r^*} & distance(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B) \le r^* \\ 0 & \text{O.W.} \end{cases} \quad \text{Equation 3.59}$$

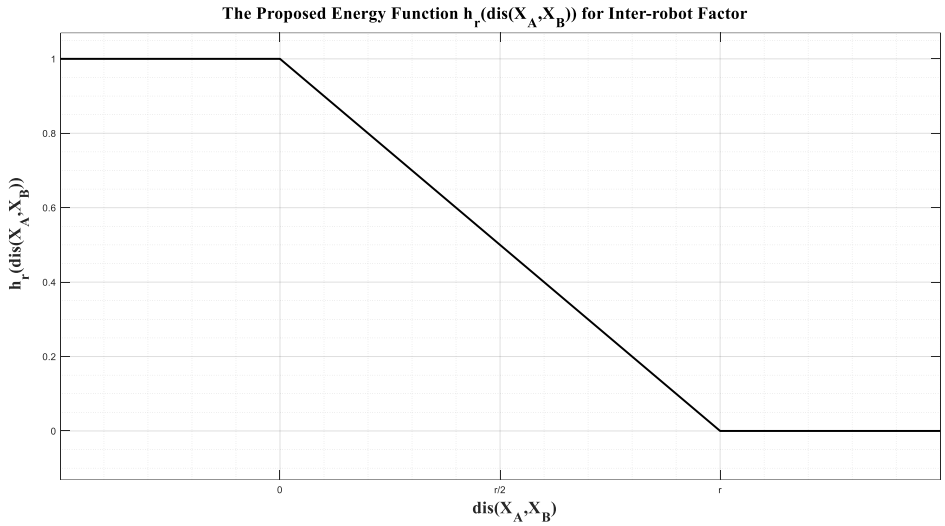


Figure 3.10 The Proposed Energy function for Inter-robot factor Equation 3.59.

About precision matrix, by this reasoning that the factor must be feebler for the nodes corresponding to further states, the [67] suggests $\boldsymbol{\Lambda}_r = (\sigma_r \times t_k)^{-2}\boldsymbol{I}_{1\times 1}$ precision matrix.

In the earlier version of [67] issued on 22 March 2022 [68], the inter-robot function energy has been proposed slightly differently. There, after introducing function $g(p)$ which is so similar to the subsequent proposed $\boldsymbol{h}_r(\boldsymbol{x}_k^A, \boldsymbol{x}_k^B)$ (that been analyzed earlier) and is a truncated Hinge loss function:

$$g(p) = \begin{cases} 1 - \dfrac{p}{r^*} & p \le r^* \\ 0 & \text{O.W.} \end{cases} \qquad \text{Equation 3.60}$$

Afterward, the proposed energy function is based on a K-point linear interpolation. Since it could be possible that the robots intersects between two consecutive timesteps, at earlier versions [68] authors decided to define energy function in the following way not to miss any kind of intersection of robots:

$$\boldsymbol{h}_r^{old}(\boldsymbol{X}_{A,i}, \boldsymbol{X}_{B,i}) = \left[g\left(r_{i+\frac{k}{K}} \right) \right]\Big|_{0\le k\le K-1} \qquad \text{Equation 3.61}$$

Where:

$$r_{i+\frac{k}{K}} = \left\| \boldsymbol{x}_{A,i} + \left(\frac{k}{K}\right)\dot{\boldsymbol{x}}_{A,i} - \boldsymbol{x}_{B,i} - \left(\frac{k}{K}\right)\dot{\boldsymbol{x}}_{B,i} \right\| \qquad \text{Equation 3.62}$$

It seems that in further versions of their studies, and based on experimental results, they found it unnecessary to implement interpolated analysis for finding intersections, and for small enough timesteps ΔT, it is not so likely to miss robots' intersections at discrete analysis.

Nonetheless, we will also use the more recent energy function (Equation 3.59). Since this energy function is not linear, we must obtain the Taylor series of this function around current variable nodes states in order to present them in conventional Gaussian form. With respect to determined variable motion states (Equation 3.48):

$$\rightarrow \boldsymbol{h}_r(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B) = \begin{cases} 1 - \dfrac{\sqrt[2]{(x_k^A - x_k^B)^2 + (y_k^A - y_k^B)^2}}{r^*} & distance(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B) \leq r^* \\ 0 & O.W. \end{cases} \quad \text{Equation 3.63}$$

$$\rightarrow distance(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B) = \sqrt[2]{(x_k^A - x_k^B)^2 + (y_k^A - y_k^B)^2} \xrightarrow{\Delta x_k^{A,B} \equiv x_k^A - x_k^B \text{ and } \Delta y_k^{A,B} \equiv y_k^A - y_k^B} \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}} \quad \text{Equation 3.64}$$

Taking the derivative with respect to positions and velocities parameters:

$$\begin{cases} \dfrac{\delta}{\delta x_k^A}\boldsymbol{h}_r = -\dfrac{\Delta x_k^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}}} \\ \dfrac{\delta}{\delta y_k^A}\boldsymbol{h}_r = -\dfrac{\Delta y_k^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}}} \\ \dfrac{\delta}{\delta x_k^B}\boldsymbol{h}_r = +\dfrac{\Delta x_k^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}}} \\ \dfrac{\delta}{\delta y_k^B}\boldsymbol{h}_r = +\dfrac{\Delta y_k^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}}} \\ \dfrac{\delta}{\delta \dot{x}_k^A}\boldsymbol{h}_r = \dfrac{\delta}{\delta \dot{y}_k^A}\boldsymbol{h}_r = \dfrac{\delta}{\delta \dot{x}_k^B}\boldsymbol{h}_r = \dfrac{\delta}{\delta \dot{y}_k^B}\boldsymbol{h}_r = \emptyset \end{cases} \quad \text{Equation 3.65}$$

Composing Jacobian matrix with respect to the obtained derivatives:

$$\boldsymbol{J}_r = \left[\frac{\delta}{\delta x_k^A}\boldsymbol{h}_r \quad \frac{\delta}{\delta y_k^A}\boldsymbol{h}_r \quad \frac{\delta}{\delta \dot{x}_k^A}\boldsymbol{h}_r \quad \frac{\delta}{\delta \dot{y}_k^A}\boldsymbol{h}_r \quad \frac{\delta}{\delta x_k^B}\boldsymbol{h}_r \quad \frac{\delta}{\delta y_k^B}\boldsymbol{h}_r \quad \frac{\delta}{\delta \dot{x}_k^B}\boldsymbol{h}_r \quad \frac{\delta}{\delta \dot{y}_k^B}\boldsymbol{h}_r\right]$$

$$\rightarrow \boldsymbol{J}_r = \frac{1}{r^* \cdot \sqrt[2]{\Delta x_k^{A,B^2} + \Delta y_k^{A,B^2}}} \times \left[-\Delta x_k^{A,B} \quad -\Delta y_k^{A,B} \quad \emptyset \quad \emptyset \quad \Delta x_k^{A,B} \quad \Delta y_k^{A,B} \quad \emptyset \quad \emptyset\right] \quad \text{Equation 3.66}$$

Now, by knowing the Jacobian matrix, the Gaussian form can simply be obtained by using Equation 3.31:

$$\rightarrow For\ dis(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B) \leq r^* \rightarrow \begin{cases} \Lambda'_r = (\sigma_r \times t_k)^{-2}(\boldsymbol{J}^\intercal \times \boldsymbol{J}) \\ \eta'_r = -(\sigma_r \times t_k)^{-2} \times \boldsymbol{J}^\intercal \times \left(1 - \dfrac{dis(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B)}{r^*}\right) \end{cases} \quad \text{Equation 3.67}$$

In Figure 3.9 An example of proposed factor graph characterization in , An example of proposed factor graph characterization in [67] is figured. In this example, there are two robots which their corresponding various types of factors are presented. In both, there are two inter-robot factor nodes between the variable nodes with distances less than the critical distance r^*. As [68] [67] claims, this excessive factor explanation aims to implement and represent the shared responsibility between two robots for doing safe planning. However, this double factor definition is a redundancy in the design that must address and solve in future works.

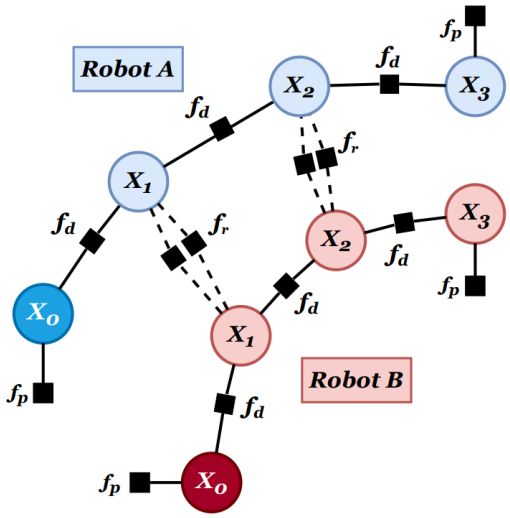


Figure 3.11 The new illustration of multi-robot planning based on factor graph characterization. [67]

d. Obstacle Factors f_o:

The concept of Obstacle factors is so like Inter-robot factors. For each variable node, there is a connected obstacle factor that describes the situation between the robot state corresponding to variable nodes and the nearest obstacle to the robot. [67] supposes some signed distance function (SDF) exists that robots can get the distance from the closest object. After obtaining the most critical distance, this distance would be passed to the obstacle factor energy function, which is almost as same as the inter-robot factor energy function, a truncated Hinge loss function:

$$\boldsymbol{h}_o(\mathbf{X}_k) = \begin{cases} 1 - \dfrac{d_o(\mathbf{X}_k)}{r_R} & d_o(\boldsymbol{x}_k) \le r_R \\ 0 & \text{otherwise} \end{cases} \qquad \text{Equation 3.68}$$

The identified precision matrix also has the frequent used form $\boldsymbol{\Lambda}_o = \sigma_o^{-2}\mathbf{I}$. Because of the considerable similarities between obstacle factor and inter-robot factor, we dismiss analyzing this factor functions specifically.

3.3.1. The Proposed Message Passing Algorithm for Gaussian Belief Propagation Planner Method

Since inter-robot factor shows its impact for $dis(\boldsymbol{X}_A^K, \boldsymbol{X}_B^K) < r^*$ and in other cases it would be ineffective, [67] makes use of a dynamic factor node assigning. As Algorithm 3.1 explains, in the beginning, after updating the starting and horizon states, The new inter-robot (or the old ones) would be created for the new robots that are coming within the defined communication range r_C (or have gotten out of the communication range). After creating (and destroying) the selected factor nodes. The Gaussian belief propagation would be done based on the strategy of two distinct message passing phases: "Internal" and "Inter-robot."

- **Internal** Message Passing:
  Includes M_I Iteration of message passing of any factors that are not connected to states of other robots. M_I could be arbitrarily large since it is an internal message passing for robots.
- **Inter-robot** Message Passing:
  Includes M_R Iteration of message passing limited to inter-robot factors associated connections. At [67] M_R is used as a bottleneck to manage the robot interactions, and this amount of iteration intentionally is selected tinier (Usually, 20% of M_I)

Algorithm 3.1 Online algorithm proposed in [67]

Algorithm 3.1: Online Planning for Robot Planning	
▪ For One Robot R_i	
1:	Update the starting and horizon states $\boldsymbol{X}_0, \boldsymbol{X}_k$ by ΔT
2:	$C(R_i)$: set of robots currently connected to R_i
	$N(R_i)$: set of robots that are laid within the communication radius (r_C) of R_i
3:	**While** Running **do**
4:	**For** $R_j \in N(R_i) \backslash C(R_i)$ **do**
5:	For newly observed robots R_j, Create inter-robot factors $f_r(R_j, R_i)$
6:	**For** $R_j \in C(R_i) \backslash N(R_i)$ **do**
7:	Delete inter-robot factors of out-of-range Robots
8:	Perform M_I Internal and M_R Inter-robot iterations to adapt the planning for the newly updated states.

3.3.2. The Visual Experimental Results of Gaussian Belief Propagation Planner Method

Although there is a set of comprehensive evaluations in [67] that investigates this method from different aspects, now we only show this method comparison with Optimal Reciprocal Collision Avoidance (ORCA) [69] that is an avoidance system. For a circle experiment for 30 existing robots. As the below figure displayed, in this complex problem, we see more smooth planning for GBP that significantly is a better trajectory than ORCA.

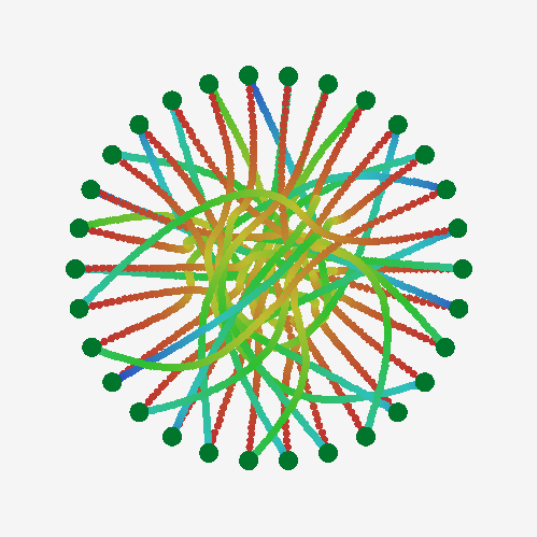


Figure 3.12 GBP planner method for circular robots' structure [67] Different colors display path planning variation via time from oldest to newest (red to blue spectrum changes).

Figure 3.13 ORCA planner method for circular robots' structure [67]. Different colors display path planning variation via time from oldest to newest (red to blue spectrum changes). In comparison with Figure 3.12, it has a longer, junkier trajectory for the same case, which worsens and diverges over time.

4 Graph-based Collision Avoidance Network Managing

As we discussed earlier around the devoted studies that addressed the available strategies for transmitting the data of cooperative perception systems, one of the few standards established by ETSI [37] can only provide the minimum required rules for cooperative transmission. It is not designed to deal with possible congestion or network overloading in V2V communications. This standard only rules some simple aspects of CPM generating and transmission, such as defining how frequently a vehicle should generate and transmit a CPM to other vehicles and which amount and types of information should be put in these messages [38]. These determined rules are fundamental that only can be used as the baselines for evaluating and investigating the performance of new proposed ruling and network managing methods.

As we discussed, one generation rule based on modifying the pre-determined ETSI rules is [39], which proposes a new dynamic CPM generation strategy. Their method is based on changing the CPM generation rate in different dynamic scenes. As illustrated in Table 2.1, it can enhance the network performance significantly only by employing their proposed dynamic policy. Detecting the redundant information associated with the detected objects, that their required information has already been sent, and preventing their transmission was another benefit that is achieved in this kind of study.

After that, we also see some context-based rulings studies that tried to propose some enhancing mechanisms in V2V applications. One of these studies is [40] which supposes having a context-based acknowledging system. At this mechanism for each broadcasting action and based on the importance and type of broadcasting content, some of the destination vehicles are deliberately asked to send back acknowledgments after receiving CPMs. [40] also shares some experimental results for this idea implementation, and due to them, in some cases, it could be possible to improve the object awareness ratio up to 82%.

Value-Anticipated V2V Communications mechanisms are another branch of the studies that attempt to make value-anticipating networks. [41] could be a good example of this group of analyses that ego vehicle tries to achieve awareness about the next CPMs. If the observed value exceeds these anticipations, this content would be selected for being written down in CPM. Speaking about value-anticipating strategy, one of the other new ideas in this study was creating a prior knowledge base in ego vehicle about the obtained CPMs in other vehicles to prioritize the transmissions order based on this prior knowledge.

Furthermore, we also see some further studies that prefer to rely on trained neural networks for managing cooperative perception systems. For example, [42] designs and trains a CNN neural network that, due to the determined reward function and reinforcement learning method, learns a suitable strategy for transmitting (or not transmitting) the generated CPMs. The training function in this study has been oriented to care about network congestion, the freshness of CPMs, and the data efficiency in identifying the new objects.

After analyzing these feasible approaches and methods, try modifying the cooperative perception systems. We thought about using the primary goal of these transmissions to propose new semantic-empowered strategies that point out clearly to this goal(s) and try to orient and manage the communications to reach this determined goal more straightforwardly. As we discussed, "collision preventing" can be selected as the main goal of this type of communication. In order to know the proposed collision detecting systems better, we have analyzed some of the proposed neural networks that get the map of the scene and LiDAR data as input and give a prediction about each detected vehicle behavior in the future [44] [45] [46] [47].

Between these bunch of studies with all different approaches, we find [47] so inspiring in support of our studies because it presents a spatially aware graph neural network, inspired by the Gaussian graphs, that tries to model the existing interactions in a traffic scene in order to predict their behaviors. Since one of the tasks of perception-expanding systems is to inform the existing vehicles about each other, this graph-based modeling has the key potential for characterizing possible situations. For example, two coop vehicles have no awareness of each other existence, and this lack of awareness could end up in some accidents. using graph-based modeling, the ego vehicle can characterize this unawareness, see its subsequent collisions, and shape its communication to avoid them.

4.1. Proposing a Factor Graph-based Collision Prediction

Based on the analyzed factor graph characterization [67]that we have studied in the previous chapter, generating the cooperative trajectories for a set of robots based on Gaussian Belief propagation is feasible. We have tried to get some inspiration from the proposed graph to design our graph architecture to anticipate the near-term collisions in a group of vehicles.

This proposed graph is not the most sophisticated tool for predicting the coming collision of a vehicle's crowd. Moreover, the initial assumption about the form of all variates' distribution, which is assumed to be Gaussian, is not entirely accurate. For example, [47] found out that in modeling the velocity and heading of detected vehicles, Von Mises distributions (the circular analogue of the normal distribution) fit better for

modeling the heading angles. Nevertheless, at rock bottom, the proposed factor graph modeling might have the potential to show us the advantages of switching from strategy-less cooperation to goal-based data exchanging. These predicted collisions would be used as one of the inputs for the policy-making function, which means these predictions would not straightly affect all data transmission planning, and this data integration with other inputs would be employed for this matter. Meaning that even for less accurate collision predictions, since these predictions would combine with more precise data to determine the cooperation strategy, this possible inaccuracy will not damage the strategies harshly.

As illustrated in the following diagram (Figure 4.1), the fundamental idea of the proposed factor graph and its repeated used factor nodes is inspired by [67]. Nevertheless, based on the intrinsic attributes of vehicle movement, some extra and modified factor nodes are needed, which are attempted to be employed in the following factor graph.

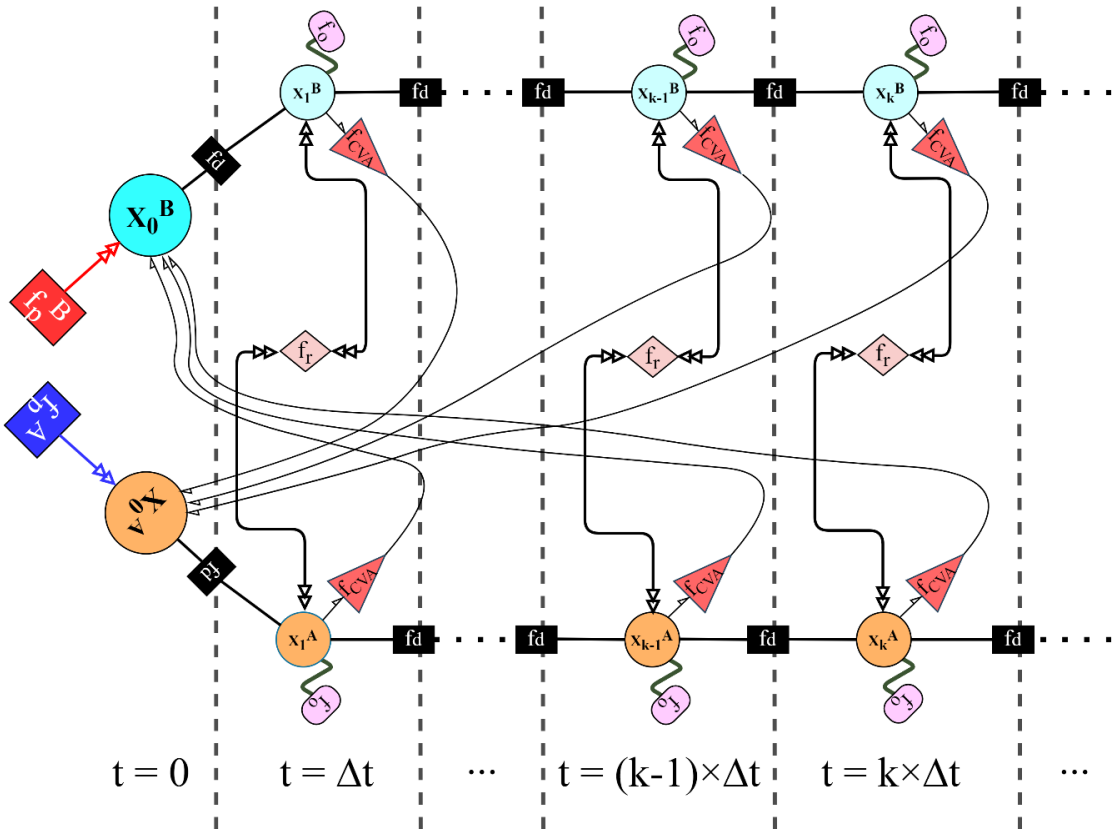


Figure 4.1 The proposed vehicles interaction factor graph composed for predicting coming collisions.

As it is illustrated, for timestep Δt, and successive analyzing intervals, we have assigned a variable node to each detected vehicle in these intervals. The identifying state for these vehicles, as same as [67], is the basic motion features of vehicles, which are position and velocity along the x and y axes. $\boldsymbol{X}_k^A = [x^A \quad y^A \quad \dot{x}^A \quad \dot{y}^A]^\top|_{t=k\Delta t}$. In addition, the exploited factor nodes are:

- **Pose Factor Node $\boldsymbol{f_p}$:**
  Which is the pose (or observation) factor node; the characterization (Λ and η parameters) of these factor nodes directly comes from the outputs of the ego vehicle perception module. The corresponding variable nodes must be created for all detected vehicles, and their present position and motion info must be implemented in the factor graph using these pose factors.
- **Dynamic Factor Node $\boldsymbol{f_d}$:**
  Implementing the dynamic factors is inspired by [67]. In our case, an additional covariance matrix realignment is being implemented for this factor node as a modification found from [68]. We will talk in more detail further about this realignment.
- **Vehicles Interaction Factor Node $\boldsymbol{f_r}$:**
  the vehicles interaction factor nodes which are used in [67] as inter-robot factor nodes. It seems there is no need for any more modifications and changes
- **Obstacle Factor Node $\boldsymbol{f_o}$:**
  The obstacle factor nodes that is obtained from [67] as well. Although here we also may follow the same meaning and approach for dealing with existing obstacles (as same as mentioned article), probably there would be some differences in detecting the obstacles in the scene, and they might no more require having a signed distance function to detect the closest (most risky) obstacle.
- **CVA Factor Node $\boldsymbol{f_{CVA}}$:**
  CVA stands for "Constant Velocity Assumption," and this factor is suggested in [68] that is being applied for simulating two different vehicles, for example, vehicles A and B. Interoperation simulation assuming each vehicles will keep moving at a constant velocity. This factor node connects to k_{th} variable node of vehicle A, $\boldsymbol{X}_k^A$, and initial variable node of vehicle B, $\boldsymbol{X}_0^B$, (and vice versa). In the following, we will explain this new factor node in more detail.

Since some of these factors have been investigated item by item in the chapter 3, there is no need to explain their energy function form and characterization further. However, it is worth inspecting the recent modification in dynamic factors and new f_{CVA} factor node.

The modification that has been done to the dynamic factor is applying a realignment matrix on the covariance matrix that reshapes this matrix and revokes its isotropic shape. This modification aims to provide more realistic interoperation scenarios, that is, non-holonomic reactions [68]. We brought up sluggishness for path swerving as one

of the intuitive principles in driving. This concept implementation is displayed in the following Figure 4.2 transcribed from [68]. In the defined intersection scenario, two robots prefer to change their speed rather than swerve their selected lanes. Evidently, applying this modification could lead to more realistic behaviors in these scenarios.

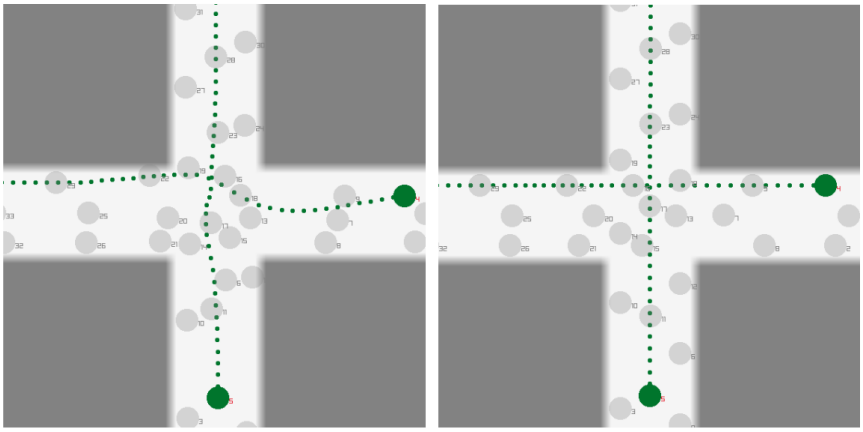


Figure 4.2 The differences between final trajectories by applying (right) and not applying (left) dynamic factor realignment (DFR). As the right figure demonstrates, by applying this alignment, robots prefer not to swerve their selected lane. [68]

4.1.1. Dynamic Factor Realignment Procedure

Remembering from dynamic factor characterization, the process noise Q_d is defined as an isotropic matrix where $Q_d = \sigma_d^2 \times \boldsymbol{I}_{2\times 2}$. This isotropic feature means that the connected states are likely to change in both x and y directions equally to avoid possible collisions. There is no elected lane (direction) for robots that prefer to stay to reach their horizon states. Realigning the process noise Q_d would make it possible to avoid collisions and accidents by accelerating or decelerating (rather than swerving). This Realigning is being done by the below equation and by implementing the transformation matrix T_{DFR}:

$$\mathbf{Q}'_d = \mathbf{T}_{DFR}\mathbf{Q}_d\mathbf{T}_{DFR}^{\top} \qquad \text{Equation 4.1}$$

The transformation matrix T_{DFR} could be calculated by knowing the preferred direction of the vehicle (which could simply be obtained by knowing the first and desired states or using the past trajectories to find the vehicle's preferred lane direction). This preferred direction is presented as a unit vector $\hat{\boldsymbol{\lambda}}$ with a corresponding orthogonal vector $\hat{\boldsymbol{\lambda}}_{\perp}$. By selecting the proper scaling parameters k_{DFR} (which mostly are less than 1 and present the dynamic factor commitment to non-holonomic steering), all required parameters are available to calculate the transformation matrix by the following Equation 4.2:

$$\mathbf{T}_{DFR} = [\hat{\boldsymbol{\lambda}} \quad k_{DFR} \quad \hat{\boldsymbol{\lambda}}_{\perp}] \qquad \text{Equation 4.2}$$

After obtaining the new process covariance matrix, the dynamic factor covariance matrix $\boldsymbol{\Sigma}_d$ is computed as same as before (Equation 3.55):

$$\boldsymbol{\Sigma}_d(\Delta t_k) = \begin{bmatrix} \frac{1}{3}\Delta t_k^3 \mathbf{Q}'_d & \frac{1}{2}\Delta t_k^2 \mathbf{Q}'_d \\ \frac{1}{2}\Delta t_k^2 \mathbf{Q}'_d & \Delta t_k \mathbf{Q}'_d \end{bmatrix} \qquad \text{Equation 4.3}$$

In Figure 4.2 Figure 4.2we have seen the significant impact of realignment on actual output. Figure 4.3 will show us the impact of transformation on covariance shape and how it would be aligned towards the horizon state by applying DRF.

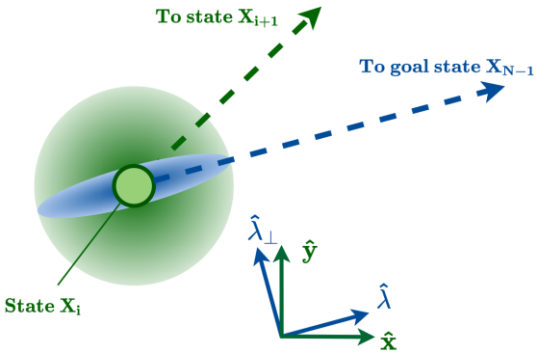


Figure 4.3 The effect of Dynamic Factor Realignment (DRF) on the shape of the process covariance of a dynamic factor. As it is illustrated, the covariance would be aligned towards the horizon state after applying DRF. [68]

4.1.2. "Constant Velocity Assumption" Factor Node

Parallel with DRF alignment, and considering that vehicles may well choose a steady direction at driving, one of the elements that helps execute this idea is the "constant velocity assumption" factor [68] that is displayed by f_{CVA} in Figure 4.1. This factor node is more or less like f_r (vehicles interaction factor node) in charge of avoiding any prospective accidents due to the vehicle estimated states at kth timestep: $\boldsymbol{X}_k^A$ and $\boldsymbol{X}_k^B$. Except that for f_{CVA} the main vehicle does not take into account the kth predicted states of another vehicle ($\boldsymbol{X}_k^B$). Instead, this factor node is connected to the head state of the second vehicle ($\boldsymbol{X}_0^B$). By assuming that vehicle B does not prefer to change its driving direction and speed, the kth state of vehicle A, $\boldsymbol{X}_k^A$, would only be changed and modified with respect to the extrapolations of the observed state $\boldsymbol{X}_0^B$.

Consequently, the CVA factor functions can be written as $f_{CVA}(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B)$ (instead of $f_{CVA}(\boldsymbol{X}_k^A, \boldsymbol{X}_k^B)$) and its corresponding energy function is:

$$\boldsymbol{h}_{CVA}(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B) = \begin{cases} 1 - \dfrac{Ex\_Polate\_dis(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B)}{r^*} & Ex\_Polate\_dis(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B) \le r^* \\ 0 & \text{O.W.} \end{cases} \qquad \text{Equation 4.4}$$

Where:

$$Ex\_Polate\_dis\left(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B\right) = \sqrt[2]{\left(x_k^A - (x_0^B + k\Delta t\dot{x}_0^B)\right)^2 + \left(y_k^A - (y_0^B + k\Delta t\dot{y}_0^B)\right)^2} \quad \text{Equation 4.5.a}$$

$$\Delta x_{Ext}^{A,B} \equiv x_k^A - (x_0^B + k\Delta t\dot{x}_0^B) \quad \text{Equation 4.5.b}$$

$$\Delta y_{Ext}^{A,B} \equiv y_k^A - (y_0^B + k\Delta t\dot{y}_0^B) \quad \text{Equation 4.5.c}$$

$$\rightarrow Ex\_Polate\_dis\left(\boldsymbol{X}_k^A, \boldsymbol{X}_0^B\right) = \sqrt[2]{\Delta x_{Ext}^{A,B\,2} + \Delta y_{Ext}^{A,B\,2}} \quad \text{Equation 4.5.d}$$

Since this energy function does not have a linear form, we must take the derivative with respect to existing parameters to obtain the Jacobian matrix for linearizing the energy function around the states $\boldsymbol{X}_k^A$ and $\boldsymbol{X}_0^B$. But we should be concerned about the extrapolation that has done, and the fact that this factor node can (and must) only affect the vehicle A state at k_{th} timestep ($\boldsymbol{X}_0^B$ is the unchangeable and fixed initial observation). Therefore, the Jacobian matrix entries corresponding to vehicle B are set equal to zero:

$$\begin{cases} \dfrac{\delta}{\delta x_k^A}\boldsymbol{h}_{CVA} = -\dfrac{\Delta x_{Ext}^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_{Ext}^{A,B\,2} + \Delta y_{Ext}^{A,B\,2}}} \\ \dfrac{\delta}{\delta y_k^A}\boldsymbol{h}_{CVA} = -\dfrac{\Delta y_{Ext}^{A,B}}{r^* \cdot \sqrt[2]{\Delta x_{Ext}^{A,B\,2} + \Delta y_{Ext}^{A,B\,2}}} \\ \dfrac{\delta}{\delta \dot{x}_k^A}\boldsymbol{h}_{CVA} = \dfrac{\delta}{\delta \dot{y}_k^A}\boldsymbol{h}_{CVA} = \emptyset \\ \dfrac{\delta}{\delta x_0^B}\boldsymbol{h}_{CVA} = \dfrac{\delta}{\delta y_0^B}\boldsymbol{h}_{CVA} = \emptyset \\ \dfrac{\delta}{\delta \dot{x}_0^B}\boldsymbol{h}_{CVA} = \dfrac{\delta}{\delta \dot{y}_0^B}\boldsymbol{h}_{CVA} = \emptyset \end{cases} \quad \text{Equation 4.6}$$

Then the composed Jacobian matrix with respect to the obtained and determined entries is:

$$\boldsymbol{J}_{CVA} = \left[\frac{\delta}{\delta x_k^A}\boldsymbol{h}_{CVA} \quad \frac{\delta}{\delta y_k^A}\boldsymbol{h}_{CVA} \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset \quad \emptyset\right] \quad \text{Equation 4.7}$$

Based on the Jacobian matrix of the CVA factor nodes, we expect a sparse matrix for the precision matrix (that would be obtained by Equation 3.31)

4.2. Strategizing the CPM Based on Vehicles Interaction Factor Graph

After introducing the vehicles interaction factor graph (illustrated in Figure 4.1), there is time to attempt to extract the valuable data from this graph to exploit them in prioritizing (or preparing) the list of the desired CPM that the ego vehicle wants to share with all, or some specific vehicles detected in its surrounding environment. For this matter, in order to have a collision-aware data transmission, we suggest the following actions be done in the ego vehicle (the source). The first is projecting its earned perception of its current environment to a factor graph. Then utilize the factor graph to predict the environment's short-term behavior (and coming dangers). Finally, using this prediction to prioritize and shape its communication strategy. These steps' detailed descriptions are stated in the following:

1. Set and identify the simulation required parameters, such as timestep Δt, number of simulation total steps K, crucial collision range r^*, communication range r_c, number of iterations per timestep M_R and M_I , and finally, setting the proper standard deviation value for node functions.
2. Ego vehicle creates the initial (current) variable states based on its perception of its surrounding environment. The initial mean vector $X_0 = [x \quad y \quad \dot{x} \quad \dot{y}]^\intercal|_{t=0}$ consists of vehicles detected position and detected (or expected) velocities for different x and y axes.
3. After defining the initial variable nodes, and based on K and Δt, the ego vehicle should compose all the remaining variable nodes for all vehicles. Therefore, implementing all the pre-defined factor nodes and connecting them to their corresponding variable nodes.

[So far, most steps are similar to the executed steps in distributing collaborative planning [67], but the main difference of execution would occur in further steps.]

4. At this step, before the message passing process, the ego vehicle must specify innovative vectors (that we call "awareness vectors") for all detected vehicles. These vectors are binary ones (including 1s and 0s) that, for the corresponding vehicle, each entry determines the vehicle's awareness of other vehicles. For example, assume that there are three vehicles in the environment:

$$\mathcal{V} = \{\mathcal{v}_1 \quad \mathcal{v}_2 \quad \mathcal{v}_3\} \qquad \text{Equation 4.8}$$

Vehicle $\mathcal{v}_1$ is assumed ego vehicle and must create awareness vectors for the other existing vehicles $\mathcal{v}_2$ and $\mathcal{v}_3$. Assume that vehicle $\mathcal{v}_1$ has recognized that, based on its perception and evaluation of the environment, $\mathcal{v}_2$ has no awareness about $\mathcal{v}_3$ and does not detect it. At the same time, $\mathcal{v}_3$ has detected all other vehicles. Furthermore, we also assume that after detecting a vehicle as

a coop vehicle by the ego vehicle, the detected vehicle would also become mutually aware of the ego vehicle.
As a result, the awareness vectors for v_2 and v_3 would be (this vector is represented by uppercase upsilon):

$$\Upsilon_{v_2} = \{1,1,0\} \quad \text{Equation 4.9.a}$$

$$\Upsilon_{v_3} = \{1,1,1\} \quad \text{Equation 4.9.b}$$

In terms of composing the awareness vectors, it so matters to employ a method in the ego vehicle for determining the awareness status between all different combinations of vehicles in a scene. We will talk more about these vectors at rest, and *we have to consider that it could be an essential topic for further studies*.

5. At this step, the ego vehicle would do the defined message passing explained in detail in Algorithm 3.1, except for the slight difference that the Iteration of message passing would be limited to both vehicle interaction and CVA factors (instead of inter-robot factors). The other significant difference is that the belief updating (Equation 3.45, Equation 3.46, and Equation 3.47) is changed here. The belief updating corresponding summations would be done with respect to the awareness **vectors**. In mathematics word, for total N vehicle in the scenes, the belief updating (Equation 3.47) for the ith vehicle would be modified to:

$$\begin{cases} \boldsymbol{\eta}_{b_i} = \boldsymbol{\eta}_{f_{obs}\to x_i} + \boldsymbol{\eta}_{f_{dyn}\to x_i} + \sum_{s\in N_{CVA}(i)} \Upsilon_{v_i} \cdot \boldsymbol{\eta}_{f_s\to x_i} + \sum_{s\in N_{Inter}(i)} \Upsilon_{v_i} \cdot \boldsymbol{\eta}_{f_s\to x_i} \\ \boldsymbol{\Lambda}_{b_i} = \boldsymbol{\Lambda}_{f_{obs}\to x_i} + \boldsymbol{\Lambda}_{f_{dyn}\to x_i} + \sum_{s\in N_{CVA}(i)} \Upsilon_{v_i} \cdot \boldsymbol{\Lambda}_{f_s\to x_i} + \sum_{s\in N_{Inter}(i)} \Upsilon_{v_i} \cdot \boldsymbol{\Lambda}_{f_s\to x_i} \end{cases} \quad \text{Equation 4.10}$$

This modification yields that the predicted behavior in the ego vehicle is based on its perception of the other vehicles' environmental awareness. Although interaction and CVA factor nodes exist between all vehicles, their passed messages are not always considered in the belief updating.

When these factor nodes do not come into the belief updating procedure, these energy functions would not also be counted in total energy minimalization. We believe that the outputs of these implemented factor nodes' function, which has the described exponential form $e^{-E(X)}$, could stay as good indicators for impending collisions. Probably the below example illustrated in Figure 4.4 could be helpful to clarify this idea.

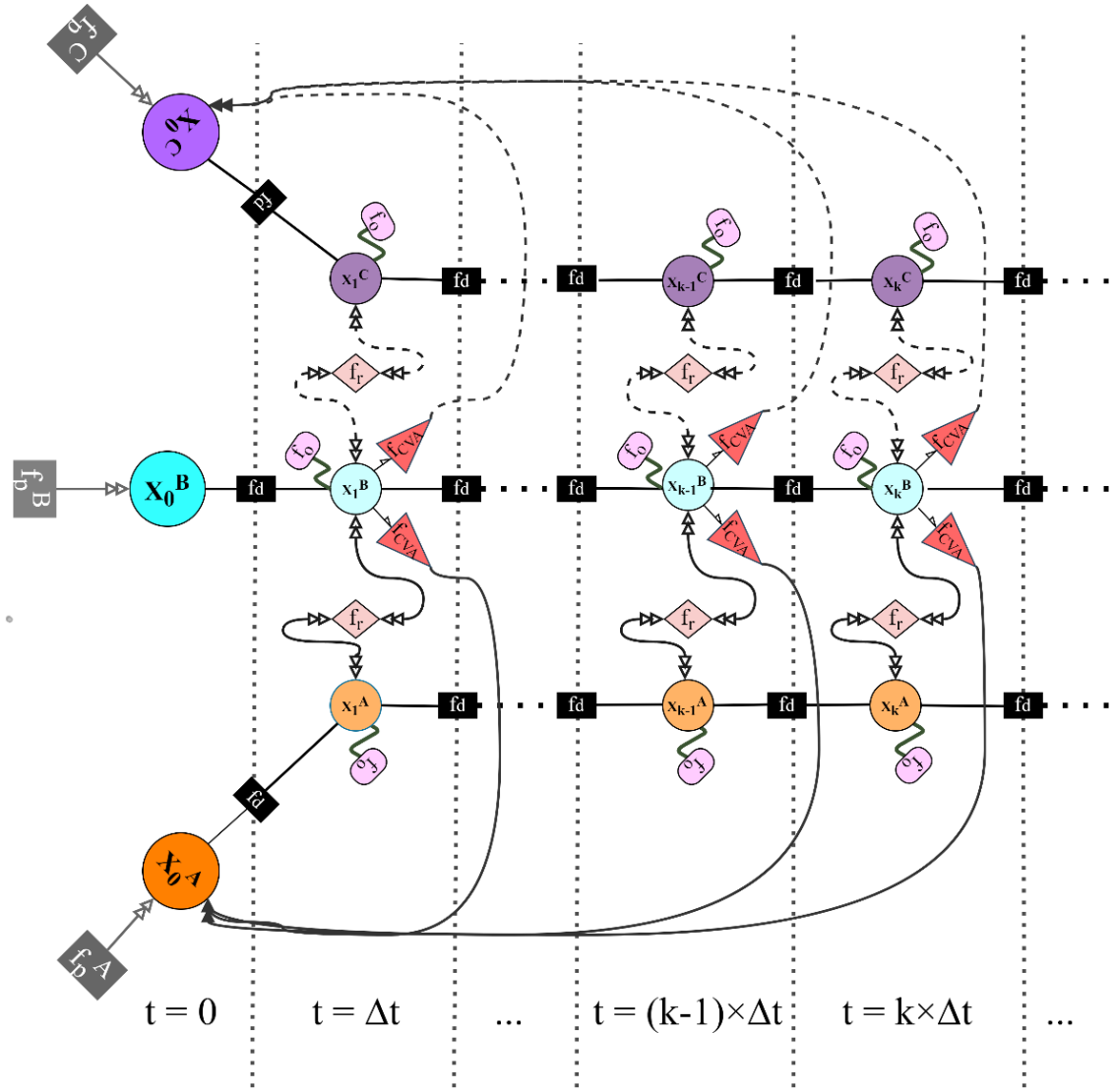


Figure 4.4 A given example for explaining vehicles interaction factor graph message passing.

For Figure 4.4, let us assume that vehicle A is an ego vehicle and has detected two other vehicles in the environment: vehicle B and vehicle C. The perceived awareness vector of vehicle B is $\Upsilon_{v_B} = [1 \quad 1 \quad 0]$, which means this vehicle has not detected vehicle C, and vehicle C influences are absent in vehicle B's future activities. Therefore, although there are installed factor nodes between vehicle B and vehicle C variable nodes, they do not affect vehicle B belief updating (This excluding, which is associated with 0 entities at the awareness vector Υ, is displayed as dotted lines in Figure 4.4). Nevertheless, in the end, vehicle A will consider the energy of these factor nodes to shape his communication strategies.

To intensify this example, assume that, based on vehicle B's direction, in the k_{th} state, it is going to collapse with vehicle C. Moreover, since it is unaware of vehicle C's existence, it will not do anything to avoid this potential collision. This predicted collision in the k_{th} state means a considerably high amount of energy for CVA and

interaction factors of vehicles B and C at the k_{th} step. As we will see, this high-level energy for interaction nodes could be used as a quality indicator of upcoming collisions that could shape transmission strategies.

In the next session, we will explain a very raw and undeveloped way to use these energy functions to select the most essential transmitting CPMs. This method is uncooked and needs more time to develop and grow. However, at least it could show a practical way to exploit the graph extracted information to manage the cooperative perception communication strategies.

4.3. Selecting the CPMs as a Combinatorial Optimization Problem

Here, we would merely introduce the fundamental idea that we believe could be effective for utilizing the anticipated data to pick up the most useful and valuable data that their transmitting could prevent potential collisions. The underlying idea is using an optimization method that, for a given total maximum value and given set of items, with corresponding weights and particular values, this method determines the best-optimized set of items concerning maximizing the collection's total value while keeping the total weight less than the affordable limit. In mathematical words, this optimization problem, which is known as Knapsack Problem, can be written as [70]:

- $I = \{1, \dots, n\}$ is a set of items.
- $S = \{s_1, \dots, s_n\}$ is the size of items I
- $W = \{w_1, \dots, w_n\}$ is the value of items I
- $C \in \mathbb{N}^+$ is the capacity of the Knapsack (or our maximum limit)

The Goal of the Knapsack Optimization Problem:

$$\arg\max_{I' \subseteq I} \sum_{i \in I'} \omega_i$$
$$s.t. \sum_{i \in I'} s_i \leq C$$

Equation 4.11

4.3.1. The Knapsack Optimization Computational Complexity

Although in the case of collaborative perception, we do not expect to have a long list of items of various sizes that make optimization a lengthy and costly procedure, it still matters to mention some of the Knapsack optimization computational complexity details.

Based on [70], for giving an instance of a the knapsack problem and a number $T \in \mathbb{N}^+$, the problem of deciding whether there is a subset I' of I with a cumulative size smaller than the capacity C and with a cumulative value of at least T is "weakly NP-hard." In

computational complexity analysis, having a weakly NP-hard problem also means that if there is a solving algorithm for this problem, it would have a polynomial running time $T(n) = O(n^k)$ with respect to problem dimensions (n), rather than the base-two logarithms of their dimensions ($T(n) = O(log_2 n)$) [71]. Therefore, there might be some solving algorithms for the Knapsack problem that can optimize it for a polynomial time of computations.

4.3.2. Customized the Knapsack Problem Concerning Our Case

The structure of the Knapsack problem seems projectable to our defined problem, for selecting the most valuable information and then writing them down in CPMs. Assuming that there is a projecting function, Ψ, that is obtained based on some determined key features. This function is designed to project a value, W, to a particular set of information (we will call it the Knapsack value function). Moreover, we need another (probably simpler) projecting function, Φ, that maps the volume or dimensions of the particular information to a number representing their size S (we will call it the Knapsack size function). By having these functions then, we can formulize the cooperative perception strategy as a Knapsack problem in this way:

1. The ego vehicle, v_i firstly determines its available exchangeable data, which could be used in cooperation perception concerning each coop vehicle as a potential for receiving its accosiated set of items:

$$\mathbb{I}_i = \bigcup_{k=1:N\backslash i} I_k \qquad \text{Equation 4.12}$$

2. After determining the list of objects, their corresponding size also should be computed and stored by using the Knapsack size function Φ:

$$\mathbb{S}_i = \bigcup_{k=1:N\backslash i} \Phi_i(I_k) \qquad \text{Equation 4.13}$$

3. For each coop vehicle $\mathcal{V} = \{v_1 \quad \cdots \quad v_N\}\backslash v_i$ and its corresponding set of items I_k, the ego vehicle, v_i determines the value of these items with respect to the vehicle state vector $\ell_k|_{k=1:N\backslash i}$, which could be a combination of features extracted from factor graphs, as well as the other essential features that could affect the communication, such as communication channel quality between ego and coop vehicle, or the prior or anticipated history knowledge of received CPMs for coop vehicles. [41]

4. After composing the vehicles' state vectors $\ell_k|_{k=1:N\setminus i}$, and arranging the exchangeable data $I_k|_{k=1:N\setminus i}$, ego vehicle uses the Knapsack value function, Ψ, to calculate the value of these sets of data, W, based on the vehicles' states:

$$\mathbb{W}_i = \bigcup_{k=1:N\setminus i} W_k$$
$$W_k = \Psi(I_k, \ell_k) \qquad \text{Equation 4.14}$$
$$\longrightarrow \mathbb{W}_i = \bigcup_{k=1:N\setminus i} \Psi(I_k, \ell_k)$$

5. After collecting and creating all essential data, and based on the available transmission capacity (and bandwidth limits) for communication with coop vehicles, there is the moment to perform the Knapsack optimization to gather the most (semantic) valuable CPMs information:

$$\textbf{For } k = 1{:}\,N \textbf{ (excluding } \boldsymbol{i}\textbf{)}$$
$$arg\,max_{I'_k \subseteq I_k} \sum_{i\in I'} W_k^i$$
$$subject\ to \sum_{i\in I'} \Phi_i\left(I_k^i\right) \le C_k \qquad \text{Equation 4.15}$$
$$\textbf{For Ends}$$

It is worth mentioning that the Knapsack optimization could also be reformulated in the following form if the ego vehicle does not have any preference for the allocated capacities of coop vehicles. Eventually, it only must fulfill the broad capacity limits.

$$\textbf{For } k = 1{:}\,N \textbf{ (excluding } \boldsymbol{i}\textbf{)}$$
$$arg\,max_{I'_k \subseteq I_k} \sum_{i\in I'} W_k^i$$
$$\textbf{For Ends} \qquad \text{Equation 4.16}$$
$$subject\ to \sum_{k=1:N\setminus i} \sum_{i\in I'} \Phi_i\left(I_k^i\right) \le C_{total}$$

Although this Idea is unprocessed, and there are a lot of sides and aspects for development, we introduce some introductory the Knapsack size and value functions (Φ and Ψ) based on created factor graphs that we can make use of them. These basic functions might not even end up with superb information collection, but they could be used as baselines for developing or proposing new Φ and Ψ functions. Regrettably, we have not found sufficient time to study and evaluate this aspect of the project in detail, and we only could propose intuitive Φ and Ψ functions regarding the though-out subjects.

- **The Knapsack Size Function $\boldsymbol{\Phi}$:**
  At first, the Knapsack size function could simply be an identity function that gives the size of each piece of information as the output.

- **The Knapsack Value Function $\boldsymbol{\Psi}$:**
  One of the possible nominees for the the Knapsack value function might be a type of sorting function that gets the output values of all fact nodes that are connected to a specific coop vehicle as the vehicles state vectors ℓ_k, and for all other coop vehicles, calculates their associating number, which shows the possibility of collision (between themself and desired coop vehicle) in the following way:

$$c_{k,j} = log\left(f_{CVA}(X_k, X_j)\right) + log\left(f_{Inter}(X_k, X_j)\right) \qquad \text{Equation 4.17}$$

  Then, for each data that is related jointly to vehicle k and j, the assigned importance would be $c_{k,j}$. Assume the set I_k is divisible by the items that are determined by their in-effect vehicles (which one of them definitely is k):

$$I_k = \bigcup i_{k,j} \qquad \text{Equation 4.18}$$

  Therefore, Ψ could be defined as:

$$\Psi(I_k, \ell_k) = \sum c_{k,j} = \sum log\left(f_{CVA}(X_k, X_j)\right) + log\left(f_{Inter}(X_k, X_j)\right) \qquad \text{Equation 4.19}$$

  This simple log-sum function can give us a number representing the importance of the set of items based on the state of connected factor nodes to each destination vehicle(s). Collecting items for notifying the coming collisions is a straightforward but effective strategy for CPMs generation that would be executed using this log-sum equation as the Knapsack value function.

5 Conclusion and Future Developments

Changing the paradigm in wireless communications and shifting from classic communication frameworks to semantic-empowered communication systems and analysing the information with respect to its semantic attributes is one of the trended topics in communication studies at this moment. We already have mentioned some influential research that proposes underlying assortment and fundamental concepts about semantic communications. Based on their achievements, we also try to concentrate our studies around V2V communication systems and find the capacities that could be thrived by applying the semantic communication ideas in these systems.

Since most of the old and even recent research that has addressed the theoretical part of semantic communication is some flawed and improper attempts to theorize semantic communications [3] [4] [5], we have seen that most other relevant studies consequently have found more practical concepts. These studies have tried to reach the promised results of semantic communications (more consistent and more extra data transmission for the same channel conditions [3]) by implementing sophisticated deep neural networks. A neural network that works as a black box but has the outstanding potential to find the proper projection that maps (and squeezes) its input data (based on the determined transmission goal) to a new encoded dimension with significantly less volume than raw input data. Besides that, we can still obtain the desired data with slight determinable distortion after transmitting and decoding this projected data. As [6] deliberates this pristine field of semantic communication in-depth, this deep learning-based tool can be considered one of the empowering tools executed in physical-bearing communication networks.

During our studies on recent articles that focus on implementing the deep learning-based encoder/decoders (and generally are published with titles including “deep semantic encoding/decoding” terms), we have observed advanced achievements in projecting the raw input data to new dimensions, which lead to massive data compression concerning expected quality withdrawing. These considerable gains in other areas of communication persuade us to steer our research around the capacities (and coming benefits) that exist in implementing semantic encoding/decoding in one of the applications-based V2V topics. During these studies, we have detected “cooperative perception” as one of the most critical applications in V2V communication. It will significantly enhance its capability if it switches from the classic Shannon’s communication framework to semantic communication.

In the cooperative perception application, most transferred data is devoted to sharing the raw sensor data captured by the ego vehicle that must be transmitted to other coop vehicles to increase their perception of the shared environment after all received data

fusion. Sharing this vast amount of raw data between moving cars to expand their perception of the environment is one of the challenges that could be treated by implementing some semantic-empowered tools. Therefore, we have dived deep into these systems to inspect them closely and find the potentials in different levels of this application that could be modified by implementing semantic-empowered modules.

For better or worse, after exploring the studies carried out around cooperative perception data sharing, we have found some impressive deep learning-based methods (and frameworks) that have addressed the challenge of sharing raw data in V2V communication systems. For example, the proposed framework in [22] divides the sharable data based on their level of processing. It analyses these different scenarios of sharing different types of gathered data (RIS, DFS, and FIS) with respect to some evaluation parameters like "fusion average precision," which is nearly a goal-oriented and semantic-empowered communication system configuration, even though the papers do not explicitly express it. Further investigation around this era also ends up finding more studies that address the issue of the vast amount of required communication sources by giving some deep learning-based approaches strikingly similar to goal-oriented methods [22] [23] [25] [26] [27] [28] [29] [30] [31].

After making deep and careful reviews of these researchers, we could not come across any part of cooperative perception application that has stayed untouched on the scale of the physical level! A large number of studies propose semantic-based solutions (unintended) for having a precise sensor fusion with less-needed data transmission. For this reason, we have decided to switch our interest slightly from the physical layer to broader scales. Then, For the rest of the thesis, we have focused on the network layer scale of V2V communications and sought modification capabilities that could be achieved by semantic empowering.

Unlike the physical layer, we have found a significant enhancement capability in the network layer by imposing the semantic features of network designing. This considerable potential in the V2V networks comes from the unsophisticated rules and standards provided for general vehicles' communications. As we discussed around the ETSI standards [37], these provided regulations can only satisfy minimum necessities, and it is not designed to deal with possible congestion or network overloading in V2V communications. Therefore, we find this era of V2V communication as an appropriate unspoilt field with a decent capacity for presenting and executing new ideas.

Although our studies around the cooperation perception strategies faced more sophisticated policies than ETSI, such as [38] [39] [40] [41] [42], we could not plainly call them semantic-empowered or goal-oriented network managing. As we argued, probably the main goal of performing this cooperative perception data exchange is to promise clearer and broader insights for vehicles to avoid any possible collisions. Routing and other applications could stand as less crucial purposes. Afterward, we

focus on the problem of how the ego vehicle may use its gathered data about the environment to detect upcoming collisions and therefore manage its data exchanging for these predictions.

In order to know the proposed collision detecting systems better, we have reviewed different vehicle behaviour predicting systems [44] [45] [46] [47] that are primarily deep neural network-based systems. As the inputs, they get the map of the scene and LiDAR data and give a prediction about each detected vehicle behaviour in the future. During this exploration, we find SpAGNN [47] as a stimulating and inspiring system that, inspired by the Gaussian graphs, composes a spatially aware graph neural network (GNN) that models the existing interactions in a traffic scene to predict their behaviours. Realizing this fact that it is possible to model the dynamic and interoperable systems (like a set of vehicles) by Gaussian graphs and predicting their short-term behaviours rings a bell in our minds that ego vehicles might use a similar method, and based on their comprehension of coop vehicles interfaces, can characterize their surrounding environment as a graph with different nodes and edges. Afterward, by executing some method (like belief propagation), predict the behaviour of the coop vehicles, and subsequently, predict the possible collisions, and based on these anticipations, shape their data exchanging.

We take advantage of the new study [67] [68] to find the proper starting point, which proposes a novel graph-based technique to address multi-robot planning problems. Their developed method (for firstly projecting a group of moving robots to a graph and then forecasting their moving based on defining dynamics and collision constraints) has a noteworthy compatibility with our uncooked idea about predicting the behaviour of a set of interoperable vehicles. Therefore, after inspecting their graph-based planning systems deliberately, we have started to build our method based on their achievements.

In our suggested goal-oriented managing method, the ego vehicle, as the source that should exchange its observed data with other vehicles, perceives the purpose beyond these data exchanges. Therefore, it does not interpret this data transmission as random and aimless. Before performing the requested data sharing, we could imagine that the ego vehicle can project its scene to a factor graph and determine the type and quality of their interactions based on these observations (and communication histories). Then, it would predict other network elements' upcomings behaviour. We have developed our idea based on the factor-graph structure presented in [67] and discussed the practicable and appropriate factor node functions that could be executed in the graph to make a more realistic behaviour prediction.

Based on our basic and light experiments, our proposed graph characterization promises to predict the upcoming dangers in a traffic scene by performing a (computationally) low-cost message passing in the equivalent factor graph. After being nearly sure about the underlying method for predicting the collisions in twisted

traffic scenes, we proposed the next phase of this goal-oriented administrator system: the optimized collision-aware CPM generation phase.

To collect the most profitable data and send them to the most needful destinations, to avoid collisions by sharing the vital data priorly, we have introduced the classic problem in combinatorial optimization known as the Knapsack problem. The main subject of this optimization is to select the most valuable items (with different weights) regarding the total weights that should not exceed the available capacity. Since we are talking about goal-oriented communication management, it should not be too hard to give value to each piece of information corresponding to its importance for reaching the main goal. Therefore, we have introduced the Knapsack value function Ψ that gets the list of available data and the coop vehicles state vectors (including the factor graph information) and gives a worth vector as the output, which determines the importance of each piece of information.

Unfortunately, due to the shortage of time, we could not analyse and assess this optimizing approach in depth to find its gains and drawbacks in the communication scenario. However, even with assuming not very encouraging results from this stage of our proposed system, we believe that we have proposed a new approach to look at the V2V communication systems, which have a great potential for modifying and enhancing the quality of V2V goal-based communication applications. A quality boost would be gained concerning the semantic importance of information. Most other related studies, in the best and most optimistic view, only consider the timely freshness or spatial age of the data as a parameter for employing specific policies in their network [42] [43], while, as we discussed, the resolute goals behind performing these data exchanges are evident and clear enough that even the network strategies can be shaped and oriented around reach these goals.

We hope further studies and carrying out the following planned developments will help us reach firm theoretical frameworks and accurate and practical systems simulations. Undeniably, proposing a new system that requires physical implementation would be pointless if some supporting simulation does not approve of its performance. So, our future planning for developing and modifying our proposed system is based on providing some touchable experiments for evaluating our suggested approaches and systems.

5.1. Future required developments

In the first phase of further development, we need to initially clarify two issues that have not been explicitly addressed in the thesis:

- The first issue is about the required interface, which relates the observation section in the ego vehicle to the communication section, and how this vehicle is supposed to translate its gathered perception to the factor graph. Although based on reviewed relevant research, most of the trained CNNs can present the detected vehicles in a scene as boundary boxes and vectors of the vehicle's dynamic feature, we need more explanation and clarifications about these two separated segments of the smart ego vehicle and their communication.

- The second unexpressed issue is about the "awareness vectors" Υ. Two challenges must be addressed in further studies about awareness vectors:
  1. The first and most crucial challenge that should be expounded is finding the precise initial awareness vector. As we discussed, the proposed method in [41] for having a history list could be helpful. Besides that, being aware of other vehicles in a scene is an ambient-based detail. Definitely, there should be some spatial analysis to compose more accurate awareness vectors. For instance, some spatial parameters, such as vehicles' mutual heading angle and their distance or obstruction between them, could play meaningful roles in the awareness measurement. Therefore, they also must be considered in the awareness vector determination.

  2. The other issue that mostly has algebraic concepts is the form of awareness vector. The assumed format for this vector, which the formulization also has done based on, is a binary format in which 0s and 1s explain the awareness of surrounding objects. While, justifiably when we talk about awareness, we should express this factor as a probability (the probability of observing, detecting, and considering), and expressing these parameters as only 0s and 1s is not so reasonable or accurate approach. Thus, one of the further developments is to reform the message passing algorithm or factor node formation to eliminate this deterministic belief updating.

After editing and correcting these stated problems in underlying aspects, further tasks are about implementing this theoretical idea in simulation space. Hopefully, we have been able to create an object-oriented program in python for creating different factor graphs and performing the Gaussian belief propagation on these graphs. In our research, these simulations were the primary technique for evaluating the proposed factor functions and belief updating algorithms. Most pieces related to variable and factor nodes have been completed till now. After completing the functions related to message passing and graph compositions, this built program would be ready to set up

and simulate more serious scenarios. In order to briefly show a part of prepared functions, the result of a simple vehicle movements prediction has been plotted below (Figure 5.1, Figure 5.2, and Figure 5.3), which indicates the feasibility of estimating the mean and covariance of vehicle positions for coming timesteps by utilizing only dynamic factors.

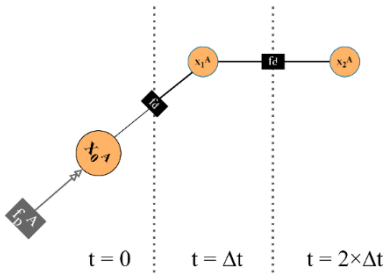


Figure 5.1 The simple factor graph that is modeled by built program

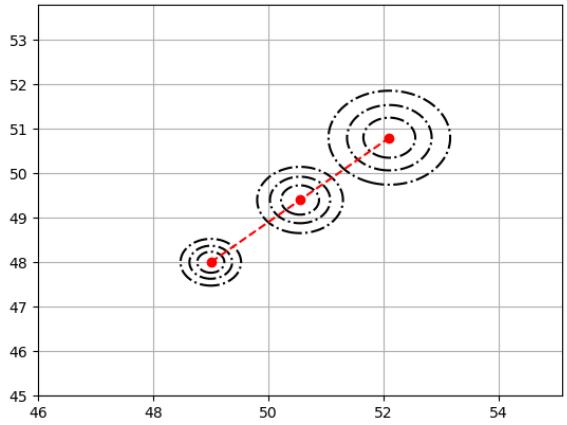


Figure 5.2 The estimated (mean and covariance) 2-D positions of a characterized vehicle in coming timesteps

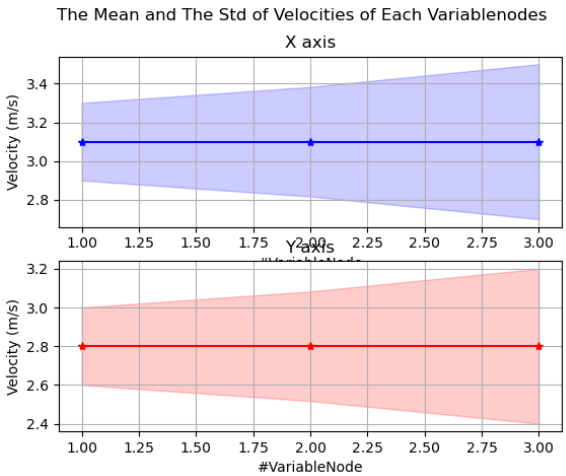


Figure 5.3 The estimated (mean and covariance) velocities of a characterized vehicle in coming timesteps

After completing the python simulation program, it is time to move to further stages of our proposed system and develop the suggested functions in the Knapsack optimization. The recommended Knapsack value function $\Psi(I, \ell)$ is too simple and only considers the limited parameters (CVA and Inter factor nodes output). Not only it would be possible to obtain more valuable data from factor graphs, but also, there is so much more vital information that could be extracted easily from the ego vehicle observations. Indeed, this information in evaluating the available exchangeable items would result in more accurate value computation.

Moreover, these factors should not be limited to the final goal of communication. They could also include more straightforward data that determines the quality of data transmissions. For example, the ego vehicle could exploit the gathered perception of its environment in a way to detect the possible quality of the channel devoted to V2V communication based on the density and obstacles placed between the source and destination point. Millimeter wave and terahertz technologies will be crucial to future V2V systems, so by applying some practical methods and algorithms (such as the mathematical modeling of [72]) for evaluating the performance of dedicated terahertz channels for portable communication sides in various traffic scenes, it could be possible to project these additional evaluations in features vector ℓ. Eventually, these developed versions of the features vector ℓ could be passed to advanced versions of the Knapsack value function $\Psi(I, \ell)$ to have a precise values assigning method regarding all aspects of the spatial factors.

Bibliography

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal,* vol. 27, no. 3, 1948.

[2] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks,* vol. 190, 2021.

[3] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland and J. A. Hendler, "Towards a theory of semantic communication," *Proceedings of the 2011 IEEE 1st International Network Science Workshop, NSW 2011,* 2011.

[4] R. Carnap, Y. Bar-Hillel and a. Others, "An outline of a theory of semantic information," *Research Laboratory of Electronics, Massachusetts Institute of Technology,* 1952.

[5] L. Floridi, "Outline of a Theory of Strongly Semantic Information," *Minds and Machines,* vol. 14, no. 2, 2004.

[6] P. Zhang, W. Xu, H. Gao, K. Niu, X. Xu, X. Qin, C. Yuan, Z. Qin, H. Zhao, J. Wei and F. Zhang, "Toward Wisdom-Evolutionary and Primitive-Concise 6G: A New Paradigm of Semantic Communication Networks," *Engineering,* vol. 8, 2022.

[7] M. Kountouris and N. Pappas, "Semantics-Empowered Communication for Networked Intelligent Systems," *IEEE Communications Magazine,* vol. 59, no. 6, 2021.

[8] J. Choi, S. W. Loke and J. Park, "A Unified Approach to Semantic Information and Communication based on Probabilistic Logic," *arXiv,* 2022.

[9] Y. Zhong, "A theory of semantic information," *China Communications,* vol. 14, pp. 1-17, 2017.

[10] A. Kolchinsky and D. H. Wolpert, "Semantic information, autonomous agency and non-equilibrium statistical physics," *Interface Focus,* vol. 8, no. 6, 2018.

[11] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine,* vol. 2, no. 2, 2007.

[12] M. Jankowski, D. Gunduz and K. Mikolajczyk, "Joint Device-Edge Inference over Wireless Links with Pruning," *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC,* Vols. 2020-May, 2020.

[13] S. Sukhbaatar, A. Szlam and R. Fergus, "Learning multiagent communication with backpropagation," *Advances in Neural Information Processing Systems,* 2016.

[14] H. A. Ameen, R. A. Zaidan, A. Mohammed, A. K. Mahamad, B. B. Zaidan, A. A. Zaidan, S. Saon, D. M. Nor, R. Q. Malik, Z. H. Kareem and S. Garfan, "A Deep Review and Analysis of Data Exchange in Vehicle-to-Vehicle Communications Systems: Coherent Taxonomy, Challenges, Motivations, Recommendations, Substantial Analysis and Future Directions," *IEEE Access,* vol. 7, 2019.

[15] Y. Wang, G. De Veciana, T. Shimizu and H. Lu, "Performance and scaling of collaborative sensing and networking for automated driving applications," *2018 IEEE International Conference on Communications Workshops, ICC Workshops 2018 - Proceedings,* 2018.

[16] H. Xie, Z. Qin, G. Y. Li and B. H. Juang, "Deep Learning Enabled Semantic Communication Systems," *IEEE Transactions on Signal Processing,* vol. 69, 2021.

[17] C. H. Lee, J. W. Lin, P. H. Chen and Y. C. Chang, "Deep Learning-Constructed Joint Transmission-Recognition for Internet of Things," *IEEE Access,* vol. 7, 2019.

[18] Z. Qin, X. Tao, J. Lu, W. Tong and G. Y. Li, "Semantic Communications: Principles and Challenges," *arXiv,* 2022.

[19] D. Huang, X. Tao, F. Gao and J. Lu, "Deep Learning-Based Image Semantic Coding for Semantic Communications," *2021 IEEE Global Communications Conference, GLOBECOM 2021 - Proceedings,* 2021.

[20] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu and G. Y. Li, "Deep Learning Enabled Semantic Communications with Speech Recognition and Synthesis," *arXiv,* 2022.

[21] X. Peng, Z. Qin, D. Huang, X. Tao, J. Lu, G. Liu and C. Pan, "A Robust Deep Learning Enabled Semantic Communication System for Text," *arXiv,* 2022.

[22] E. E. Marvasti, A. Raftari, A. E. Marvasti, Y. P. Fallah, R. Guo and H. Lu, "Feature Sharing and Integration for Cooperative Cognition and Perception with Volumetric Sensors," *arXiv,* 2020.

[23] Q. Chen, S. Tang, Q. Yang and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," *Proceedings - International Conference on Distributed Computing Systems,* Vols. 2019-July, 2019.

[24] S. M. Gani, Y. P. Fallah, G. Bansal and T. Shimizu, "Message content control for distributed map sharing in vehicle safety communications," *2017 IEEE 36th International Performance Computing and Communications Conference, IPCCC 2017,* Vols. 2018-January, 2018.

[25] M. Fanaei, A. Tahmasbi-Sarvestani, Y. P. Fallah, G. Bansal, M. C. Valenti and J. B. Kenney, "Adaptive content control for communication amongst cooperative automated vehicles," *2014 IEEE 6th International Symposium on Wireless Vehicular Communications, WiVeC 2014 - Proceedings,* 2014.

[26] E. E. Marvasti, A. Raftari, A. E. Marvasti, Y. P. Fallah, R. Guo and H. Lu, "Cooperative LIDAR Object Detection via Feature Sharing in Deep Networks," *IEEE Vehicular Technology Conference,* Vols. 2020-November, 2020.

[27] E. E. Marvasti, A. Raftari, A. E. Marvasti and Y. P. Fallah, "Bandwidth-adaptive feature sharing for cooperative LIDAR object detection," *2020 IEEE 3rd Connected and Automated Vehicles Symposium, CAVS 2020 - Proceedings,* 2020.

[28] A. Rauch, F. Klanner, R. Rasshofer and K. Dietmayer, "Car2X-based perception in a high-level fusion architecture for cooperative perception systems," *IEEE Intelligent Vehicles Symposium, Proceedings,* 2012.

[29] W. Liu, S. W. Kim, Z. J. Chong, X. T. Shen and M. H. Ang, "Motion planning using cooperative perception on urban road," *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM),* pp. 130-137, 2013.

[30] A. Rauch, S. Maier, F. Klanner and K. Dietmayer, "Inter-vehicle object association for cooperative perception systems," *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013),* pp. 893-898, 2013.

[31] 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," *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, SEC 2019,* 2019.

[32] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Vols. 2016-December, 2016.

[33] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2018.

[34] S. Malik, M. A. Khan and H. El-Sayed, "CARLA: Car Learning to Act - An Inside Out," *Procedia Computer Science,* vol. 198, 2021.

[35] 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," *arXiv,* 2021.

[36] H. Noh, S. Hong and B. Han, "Learning Deconvolution Network for Semantic Segmentation," *arXiv,* 2015.

[37] "Intelligent Transport Systems (ITS); Vehicular Communications;Basic Set of Applications; Analysis of the Collective Perception Service (CPS)," ETSI TR 103 562 V2.1.1, 2019.

[38] T. Gokulnath, S. Miguel and G. Javier, "Generation of Cooperative Perception Messages for Connected and Automated Vehicles," *{IEEE} Transactions on Vehicular Technology* , vol. 69, pp. 16336-16341, 2020.

[39] G. Thandavarayan, M. Sepulcre and J. Gozalvez, "Analysis of message generation rules for collective perception in connected and automated driving," *IEEE Intelligent Vehicles Symposium, Proceedings,* Vols. 2019-June, 2019.

[40] B. Coll-Perales, G. Thandavarayan, M. Sepulcre and J. Gozalvez, "Context-based Broadcast Acknowledgement for Enhanced Reliability of Cooperative V2X Messages," *2020 Forum on Integrated and Sustainable Transportation Systems, FISTS 2020,* 2020.

[41] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi and O. Altintas, "Value-anticipating V2V communications for cooperative perception," *IEEE Intelligent Vehicles Symposium, Proceedings,* Vols. 2019-June, 2019.

[42] S. Aoki, T. Higuchi and O. Altintas, "Cooperative Perception with Deep Reinforcement Learning for Connected Vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings,* 2020.

[43] J. A. Rahal, G. D. Veciana, T. Shimizu and H. Lu, "Optimizing Timely Coverage in Communication Constrained Collaborative Sensing Systems," *2020 18th*

International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOPT 2020, 2020.

[44] S. Casas, W. Luo and R. Urtasun, "IntentNet: Learning to Predict Intention from Raw Sensor Data," *Proceedings of The 2nd Conference on Robot Learning* , vol. 87, pp. 947-956, 2018.

[45] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas and R. Urtasun, "End-to-end interpretable neural motion planner," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Vols. 2019-June, 2019.

[46] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi and S. Savarese, "CAR-Net: Clairvoyant Attentive Recurrent Network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 11215 LNCS, 2018.

[47] S. Casas, C. Gulino, R. Liao and R. Urtasun, "SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data," *Proceedings - IEEE International Conference on Robotics and Automation,* 2020.

[48] B. Yang, W. Luo and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2018.

[49] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks,* vol. 20, no. 1, 2009.

[50] "Autophagy related 4D," Ma'ayan Laboratory of Computational Systems Biology, [Online]. Available: https://maayanlab.cloud/Harmonizome/gene/ATG4D.

[51] H.-A. Loeliger, "Factor Graphs and Message Passing Algorithms — Part 1: Introduction," Swiss Federal Institute of Technology Zurich .

[52] D. Wisth, M. Camurri and M. Fallon, "Robust Legged Robot State Estimation Using Factor Graph Optimization," 2019. [Online].

[53] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," 2017. [Online].

[54] J. Ortiz, T. Evans and A. . J. Davison, "A visual introduction to Gaussian Belief Propagation," 21 June 2021. [Online]. Available: gaussianbp.github.io.

[55] "Factor graph," [Online]. Available: en.wikipedia.org/wiki/Factor_graph.

[56] H. E. Kyburg and J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.," 1991. [Online].

[57] K. Murphy, Y. Weiss and M. I. Jordan, "Loopy Belief Propagation for Approximate Inference: An Empirical Study," *arXiv,* 2013.

[58] J. S. Yedidia, W. Freeman and Y. Weiss, "Generalized Belief Propagation," in *Advances in Neural Information Processing Systems*, MIT Press, 2000.

[59] Y. Weiss and W. T. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," 2001. [Online].

[60] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," 2008. [Online].

[61] O. Meshi, A. Jaimovich, A. Globerson and N. Friedman, "Convexifying the Bethe free energy," 2009. [Online].

[62] D. Bickson, "Gaussian Belief Propagation: Theory and Aplication," 2008.

[63] J. Du, S. Ma, Y.-C. Wu, S. Kar and J. M. F. Moura, "Convergence Analysis of Belief Propagation on Gaussian Graphical Models," *arXiv,* 2018.

[64] Q. Su and Y. C. Wu, "On convergence conditions of Gaussian belief propagation," 2015. [Online].

[65] A. J. Davison and J. Ortiz, "FutureMapping 2: Gaussian Belief Propagation for Spatial AI," *arXiv,* 2019.

[66] "Pattern Recognition and Machine Learning," 2007. [Online].

[67] A. Patwardhan, R. Murai and A. J. Davison, "Distributing Collaborative Multi-Robot Planning with Gaussian Belief Propagation," *arXiv,* 2022.

[68] A. Patwardhan, R. Murai and A. J. Davison, Distributing Collaborative Multi-Robot Planning with Gaussian Belief Propagation - Version 1, arXiv, 2022/03/22.

[69] J. Van Den Berg, S. J. Guy, M. Lin and D. Manocha, "Reciprocal n-body collision avoidance," 2011. [Online].

[70] D. Connolly, S. Martello and P. Toth, "Knapsack Problems: Algorithms and Computer Implementations," 1991. [Online].

[71] M. Garey and D. S. Johnson, Computers and Intractability, W. H. Freeman and Company, 1979.

[72] D. Moltchanov, E. Sopin, V. Begishev, A. Samuylov, Y. Koucheryavy and K. Samouylov, "A Tutorial on Mathematical Modeling of 5G/6G Millimeter Wave and Terahertz Cellular Systems," *IEEE Communications Surveys & Tutorials,* vol. 24, pp. 1072-1116, 2022.

[73] Prometeus, "Fire". Olympus Patent 1, 300.000 BC.

[74] G. Thandavarayan, M. Sepulcre and J. Gozalvez, "Generation of Cooperative Perception Messages for Connected and Automated Vehicles," 2020. [Online].

[75] D. Pisinger, H. Kellerer and U. Pferschy, Knapsack Problems, Springer, 2004.

List of Figures

List of Tables