Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

PhD Thesis

Pheromone-Enhanced Reinforcement Learning for Multi-Agent Self Organization in Partially Observable Non-Stationary Environments

Author:
Hriday Nilesh Sanghvi

Supervisors:
Prof. Vinny Cahill
Prof. Ivana Dusparic

17th June 2025

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed:

_____

Hriday Nilesh Sanghvi, 31st March 2024

Abstract

A complex system refers to a collection of interconnected components or agents that interact with each other in nonlinear ways. Reinforcement learning (RL) is an optimization technique where an agent learns to make decisions through feedback signals obtained by interacting with its environment. It is well-suited for real-world problems as it can learn without needing a model of the environment. Although, as the number of agents in a system grows, centralized control with RL becomes impractical due to scalability limitations.

This scalability issue prompts the need for decentralized control with multi-agent reinforcement learning (MARL). However, MARL faces a deep-rooted challenge in coordinating actions of multiple agents to achieve common objectives in partially observable non-stationary environments. Some complex multi-agent systems (MAS) in nature like ant colonies, bee swarms or bird flocks can self-organize and exhibit emergent coordinated behaviour. For example, during foraging of food, ants can emit chemical pheromones that serve as an indirect form of communication (known as stigmergy) and influence the movement of other ants perceiving this chemical, displaying collective swarm behaviour.

To address the challenge of coordination in MARL for partially observable and non-stationary environments, this thesis proposes Pheromone-Enhanced Reinforcement Learning (PERL), an approach that adopts stigmergic communication mechanisms from Swarm Intelligence (SI) methods like Ant Colony Optimization (ACO). PERL integrates digital pheromones and stigmergy with MARL. Stigmergic mechanisms, like evaporation, ensure the assimilation of new information, while diffusion facilitates the transmission of pheromones throughout the entire system, promoting self-organization at a systemic level. This results in emergent behavior arising from individual agent interactions, enabling coordination amidst partially observable and non-stationary conditions.

To tackle the challenges of a real-world complex system characterized by partial observ-

ability and non-stationarity, demanding effective coordination, we conducted evaluations, we evaluated PERL in managing traffic with Connected and Autonomous Vehicles (CAVs). Our assessment covered a range of traffic scenarios, including multi-CAV navigation involving lane changes amidst dynamic obstacles on multilane highways, as well as the negotiation of single and interconnected four-way intersections with conflicting unidirectional unilane flows. The diversity of tested applications demonstrates that PERL is application-agnostic.

Our baselines include two distinct rule-based or heuristic approaches, and MARL without the use of communication strategies (referred to as Independent RL or IRL). Empirical studies demonstrates that PERL outperforms all baseline approaches. The time taken for all agents to complete the simulation and mean waiting time is better than baselines by atleast $\sim 13\%$ and $\sim 24\%$, respectively, in the intersection scenario. Additionally, PERL's mean recovery time after perturbations is also better by atleast $\sim 18\%$ in highway lane-changing scenarios with dynamic blockages.

The feedback signal proposed by PERL incentivizes minimizing deviation in local pheromone levels. Empirical results show that this translates to an objective to achieve a system-wide equitable distribution of pheromones, through stigmergic mechanisms integrated into PERL, and emerges as a self-organizing behavior that aims to redistribute traffic, which is seen to improve overall traffic flow. Furthermore, the evaluation results demonstrate that PERL exhibits a tendency to approach global pheromone equilibrium $\sim$20% earlier than its baseline counterparts.

Acknowledgements

I express heartfelt appreciation to my supervisor, Prof. Vinny Cahill, and co-supervisor, Prof. Ivana Dusparic, for their invaluable guidance, profound insight, and remarkable patience. Without their unwavering support, I would not have successfully navigated this journey. Their expertise, mentorship, and commitment to excellence have significantly enriched my academic and personal growth.

I would also like to extend my gratitude to Dr. Lara Codeca, a former member of the DSG, for her support from the beginning, which continued halfway through my PhD studies. I am also deeply thankful to my fellow researchers and friends for reminding me that I am not alone in my struggles.

Last but not least, I am truly grateful to my parents for their unconditional love, unwavering belief in my abilities, and full support throughout this journey. Their sacrifices go above and beyond the call of duty as parents.

This work would not have been possible without the collective contributions and support of these individuals, and for that, I am eternally grateful.

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

Contents

List of Figures

List of Tables

1 Introduction

This thesis explores the fusion of swarm intelligence with multi-agent reinforcement learning (MARL) to instigate self-organization (SO) [9] within multi-agent systems (MAS). SO (discussed in Section 2.4) is the spontaneous emergence of order, structure, pattern, or behavior within a complex system without requiring external intervention or explicit behavior definitions.

The proposed approach, Pheromone-Enhanced Reinforcement Learning (PERL), integrates the indirect communication capability of stigmergy in ACO with MARL, enabling MARL to learn policies that can effectively coordinate in partially observable and non-stationary environments. PERL does not require any global information, since the actions performed by local agents gives rise to emergent coordinated behaviour through SO. We evaluate PERL in the context of traffic management, a prime example of real-world problem exhibiting partial observability and non-stationarity. It has demonstrated its superiority over traditional rule-based approaches and MARL that lacks communication capabilities in multi-agent scenarios (Independent RL or IRL).

This chapter serves as an introductory overview of the research, delving into fundamental concepts within Swarm Intelligence (SI) [10], notably stigmergy and Ant Colony Optimization (ACO) [11, 12], challenges in MARL, as well as introducing our proposed approach, PERL. Furthermore, it outlines the primary contributions of the thesis and presents a roadmap for the subsequent chapters, drawing upon any information used in the confirmation report [13].

1.1 Limitations of MARL

The nature of MARL is complex and dynamic due to continuously changing states and strategies of multiple entities, and as such, it brings with it a whole new set of challenges such as non-stationarity and partial observability (explained further in chapter 2). Essentially, partial observability refers to situations where agents have incomplete information about the environment or the state of other agents within the system. On the other hand, non-stationarity refers to environments where the underlying dynamics or conditions change over time. These changes can occur due to various factors such as shifting traffic patterns, evolving weather conditions, or the introduction of new obstacles or blockages. In partially observable non-stationary environments, coordination in MARL becomes challenging since agents must adapt to changing conditions and make decisions based on incomplete information.

In addition to those discussed, MARL faces further challenges such as the curse of dimensionality, sparse feedback signals, consensus issues, and emergent complexity (see Chapter 2). The "curse of dimensionality" refers to the exponential rise in computational demands as the state or action space expands, often necessitating decentralized control in MARL systems. Sparse feedback signals mean agents receive limited or infrequent feedback on their actions' outcomes. Consensus presents the difficulty of reaching agreement on optimal actions among multiple agents with conflicting objectives. Emergent complexity describes the phenomenon of system-level behavior emerging or arising from simple individual agent interactions.

To tackle the challenges posed by partial observability and non-stationarity, our approach draws inspiration from Swarm Intelligence (SI) principles, notably communication and behavioral feedback using digital pheromones akin to Ant Colony Optimization (ACO) (discussed in Chapter 2). PERL leverages the phenomena of emergence in complex systems to enhance coordination between self-organizing MARL agents. However, PERL in this thesis, does not directly address the consensus limitation inherent in MARL.

1.2 Problem Statement

A longstanding challenge in MARL lies in addressing the lack of coordination between agents interacting in partially observable [14–16] and non-stationary [17–19] environments, particularly at scale. Centralized MARL [20–23] lacks scalablity with the number of agents in the system. Hence, we draw inspiration from decentralized multi-agent systems in nature like ant colonies or bee swarms or bird flocks. These complex systems exhibit self-organization that leads to emergent behaviour. In ACO (see section 2.5.1) for example, ants use stigmergy as an indirect form of communication to influence swarm behaviour. SI techniques such as stigmergy in ACO represent a viable indirect communication mechanism capable of fostering coordination within partially observable and non-stationary environments. One pertinent real-world problem that vividly exhibits non-stationarity and partial observability within its environmental dynamics, thereby emphasizing the critical need for effective coordination is traffic management.

1.3 Research Question

How can stigmergic communication mechanisms, such as Ant Colony Optimization (ACO) from Swarm Intelligence (SI), enhance coordination in Multi-Agent Reinforcement Learning (MARL) for environments characterized by partial observability and non-stationarity? Furthermore, how efficiently can emergent behaviors resulting from local interactions coordinate the actions of multiple agents to achieve system-wide objectives in real-world application domains?

1.4 Hypothesis

We hypothesize that integrating stigmergy principles from ACO with MARL will induce self-organization, resulting in emergent coordinated behavior in partially observable and non-stationary environments. Specifically, our approach, PERL, can achieve this by integrating digital pheromones and stigmergic processes into the state representation of MARL agents. Alongside a novel positive feedback signal promoting minimal deviation in locally

perceived pheromone levels, this integration facilitates emergent coordination among CAVs. Through mechanisms such as evaporation and diffusion, the feedback from local interactions is expected to influence global objectives, fostering self-organizing behavior aimed at redistributing traffic and ultimately enhancing traffic flow in the system.

1.5 Pheromone-Enhanced Reinforcement Learning (PERL)

Our innovation, Pheromone-enhanced Reinforcement Learning (PERL), represents a significant departure from traditional approaches. At its core, PERL leverages the concept of stigmergy, wherein agents communicate indirectly through environmental signals rather than direct communication. In our context, vehicles are assumed to utilize vehicular ad hoc network (VANET) technology to sense and respond to pheromone signals left behind by their counterparts.

Unlike conventional methods that rely on storing and processing sequential data frames, the pheromone representation of the environment offers a dynamic and simplified snapshot of the state. This pheromone-enhanced state, continually updates in response to changes in the environment. Processes such as evaporation play a crucial role in PERL by gradually removing old or outdated information from the pheromone field. This mechanism helps prevent the accumulation of irrelevant data, ensuring that agents focus on the most relevant and recent information available.

The evaporation factor ranges between 0 and 1 and is a representation of how much of the old information will disappear. For example, if the evaporation factor is set to 0.3, then 30% of the old data will vanish.

Similarly, diffusion facilitates the spread of information from local areas to a more global scale within the environment. This ensures that individual agent experiences are shared and utilized effectively across the entire system, enhancing global coordination.

The diffusion factor also ranges between 0 and 1 and is a representation of how much of the information contained locally will spread to surrounding areas. For example, if diffusion factor is set to 0.7, then 70% of the local information will spread.

In our implementation, we adopt a strategy of discretizing (see Chapter 4) the pheromone

field into *zones*. This approach serves to minimize computational overhead [24] and maintain simplicity and interpretability of the pheromone field values during evaluation. This enables us to effectively investigate the core concepts and principles underlying our hypothesis, while mitigating potential computational challenges that may arise from attempting to model complex, continuous environments.

1.6 PERL in Traffic

An extensively visible phenomenon in road networks today is traffic congestion. Characterized by delays in travel time, road rage and accidents, increased fuel consumption and carbon emissions, and longer waiting times [25], congestion is widespread on a global scale [26]. There are two main categories of congestion - recurring congestion and nonrecurring congestion. Recurring congestion is the expected congestion that occurs regularly, due to vehicle density overwhelming the road capacity due to inadequate road provisioning or sub-optimal technology to manage dynamic traffic flow; nonrecurring congestion includes congestion caused by special events like concerts or sports events, construction work, and other unpredictable events like the weather or road incidents [25]. Research indicates recurring congestion forms only 45% of congestion, while nonrecurring congestion is responsible for 55% of the overall congestion [27] (Figure 1.1).

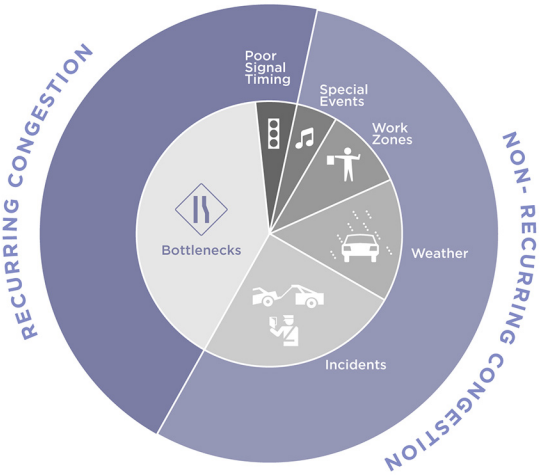

Figure 1.1: Sources of congestion as detailed in [1]

Minimization of congestion is essentially an optimization problem. Reinforcement Learning (RL) as discussed further in chapter 2, a behavioural learning methodology based on

trial-and-error in an environment that provides feedback, is known to work well for optimization problems [28]. Existing traffic flow optimization methods include intersection-centric methods, both using centralized [20–23, 29–33] and decentralized architectures [34–40]. Intersection-centric approaches usually refer to Urban Traffic Control (UTC) methods. The central point of control in independent intersections is a junction controller that only relies on local junction information and does not take into account any data from the other intersections in the system. In coordinated intersections, a centralized control approach like the Split cycle offset optimisation technique (SCOOT) [41], explained further in chapter 2), optimizes signal timings for coordination of multiple intersections with the data from all of them. Meanwhile, in the decentralized approach, every intersection has its own junction controller for decision making, which also considers data from other intersections to best coordinate system-wide traffic flow. However, UTC is more adept at dealing with the predictable (recurring) side of traffic congestion and often finds it difficult to quickly recover from unpredictable changes in the system that cause congestion. Running individual single-agent RL algorithms in a distributed manner for each vehicle-agent does not guarantee a convergence to a system-level objective [35] and hence will not bring about system-wide level of coordination. With the entry of CAVs, there was also some work based on the coupled control of intersections and CAVs [42–44], including platooning work [45–48]. This type of infrastructure-centric system that can exercise control over individual vehicles would be quite useful for the transitionary period from the reliance on traffic-light control to fully decentralized CAVs depending on their penetration rate. As the adoption of CAVs in the real world increases in the coming years, the natural course of action would be to replicate the successful architectures in UTC for the new and improved CAV-based data in real-time. The extrapolation of centralized approach used in UTC, however, will not scale with the number of vehicles in real-world scenarios due to the curse of dimensionality (discussed in Chapter 2). This encourages the possibility of a distributed control approach to the vehicle-centric coordination problem.

The advanced communication technology found in CAVs enables much more accurate sensing of data that can be used to generate an accurate traffic state representation [49–51], and the control of traffic flow could also be made more fine-grained and precise by virtue of

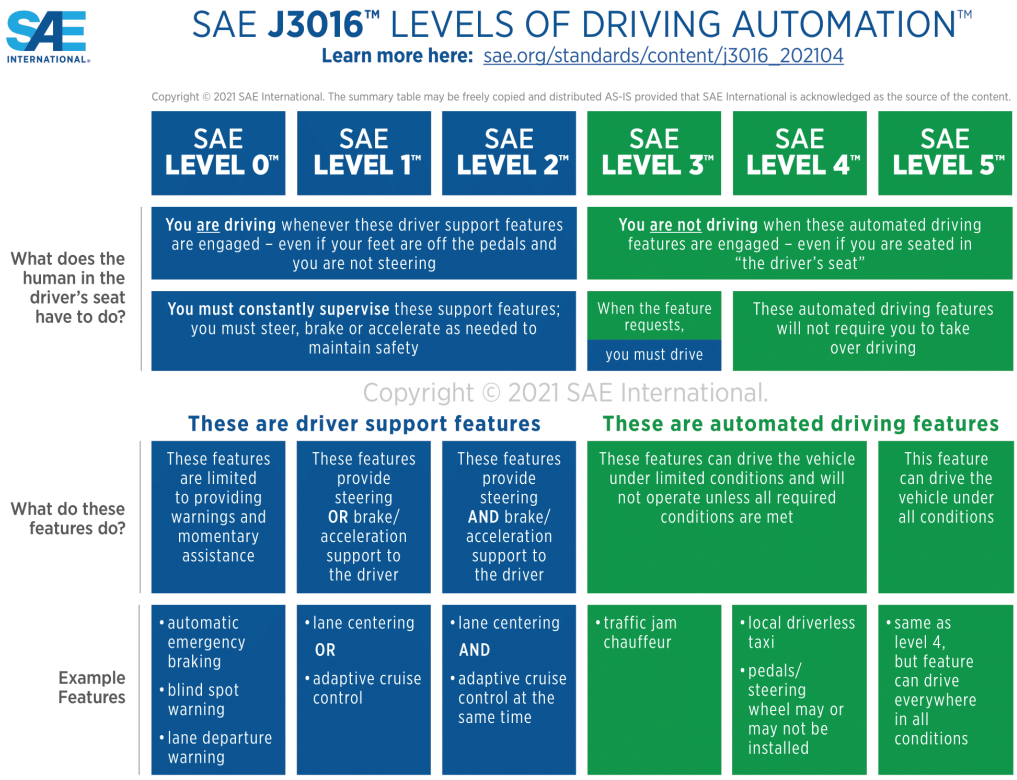

Figure 1.2: Levels of autonomy detailed in [2]

autonomy of individual vehicles in real-time. An intuitively simple approach to coordinating CAVs in a multi-agent setting can be derived from nature itself [9]. To do this, we must conceptualize this system as a system that organizes itself without any external control. Formally, it is referred to as a self-organizing system [52]. Due to the flexible re-structuring seen in self-organizing systems, it is proposed to be much more effective in tackling the larger non-recurring congestion problem. In this thesis, efforts are focused on evaluating PERL in the context of traffic flow optimization, primarily during non-recurring congestion scenarios.

Hence, we evaluate the effectiveness of PERL in multilane highways with dynamic blockages, negotiation at single intersection and multiple connected intersections.

1.7 Thesis Assumptions

When designing and assessing PERL, this thesis makes several assumptions regarding the deployment environment. These design assumptions serve to delineate the scope of the thesis by defining the specific issues that PERL aims to address. Simultaneously, the assumptions about the evaluation environment are dictated by the capabilities and constraints of the simulation testbed utilized.

1. **Failure-Free Agents:** In our research, we assume that the agents, representing entities within the system, operate without encountering failures. This means that they are capable of maintaining consistent communication with their neighboring agents and reliably receiving accurate information throughout the simulation. By assuming failure-free agents, we can focus our analysis on the behavior and interactions of the agents themselves, without having to account for potential disruptions or communication breakdowns that may occur in real-world scenarios.
2. **Homogeneous Agents:** All agents in our system are assumed to be homogeneous, meaning they share identical characteristics and capabilities. This includes uniform acceleration and deceleration capabilities, ensuring that all agents behave in a consistent manner. By assuming homogeneity among agents, we can simplify the modeling and analysis process, as we do not need to account for variations in behavior or capabilities among different agent types. This allows us to focus on understanding the collective behavior of the system as a whole.
3. **Synchronized System Time:** We assume that system time is synchronized across all agents in our simulations. This means that all actions and interactions within the system occur simultaneously at fixed time intervals or multiples thereof. By synchronizing system time, we can ensure that agents make decisions and execute actions in parallel, facilitating coordinated behavior and interactions. This synchronized timing also allows for more efficient simulation and analysis of the system dynamics, as it eliminates potential discrepancies in the timing of events between different agents.
4. **Absence of Consensus Consideration in PERL:** Since PERL relies heavily on emergent coordination through self-organization, PERL in this thesis, not directly

address the limitation of the lack of consensus building in MARL.

5. **Simple grid-based networks:** The experiments in this thesis are conducted exclusively on simple grid-based networks. This choice allows for a controlled environment to isolate and analyze the effects of self-organization and MARL without additional complexities introduced by real-world topologies. However, this presents a potential limitation, as real-world road networks exhibit irregular structures. The implications of this limitation should be considered when extrapolating the results to practical traffic scenarios.

Applying PERL to more realistic traffic scenarios would require relaxing several key assumptions made in this study. The following challenges must be addressed for such an extension:

- **Network Complexity:** Real-world road networks have varying intersection types, irregular road lengths, and asymmetric lane structures, which may require additional modifications to the environment representation in PERL.
- **Heterogeneous Traffic:** Unlike the homogeneous agents in these grid-based networks, real-world traffic consists of diverse vehicle types with different acceleration profiles and route preferences, requiring more sophisticated learning dynamics.
- **Scalability Concerns:** Larger, real-world networks introduce computational challenges, as MARL algorithms must efficiently scale to thousands of interacting agents. This may require distributed computing techniques.
- **Realistic Data Integration:** Extending PERL to real-world networks would necessitate integrating real-world traffic data (e.g., from city traffic sensors) to ensure policy learning remains applicable in real deployment.

1.8 Contributions

- **Development of Conceptual Framework for integrating stigmergy with MARL:** This research advances the conceptual framework by integrating principles from swarm intelligence with MARL. Specifically, it implements methodologies for merging stigmergic mechanisms, inspired by ACO, with MARL to address coordina-

tion challenges within partially observable non-stationary multi-agent systems.

- **Integration of Digital Pheromones with MARL:** The research explores different ways of integrating digital pheromones with MARL, including the embedding of pheromone within the state representation of the MARL agent, adding pheromone-based exploration noise or direct action noise, and pheromone-based feedback signals.
- **Novel feedback signal for system-wide objective:** Our research introduces a novel pheromone-based feedback signal incentivizing minimal deviation of local pheromone levels. This signal is designed to propagate throughout the system, promoting global equalization in pheromone levels through stigmergy. By encouraging equalized traffic redistribution, the system tends to converge towards a state *consistent* with equilibrium-like behavior [53], which we regard as an attractor state for self-organizing systems. This redistribution towards equilibrium-like states has been theoretically associated with improved traffic flow [54].
- **Features for Pheromone Calculation:** This research introduces application-specific features for the calculation of pheromones, and their computation, requiring understanding of the application domain.
- **Empirical Evaluation and Insights:** Empirical evaluations conducted across various traffic scenarios demonstrate the effectiveness of PERL in improving coordination, establishing that PERL is application agnostic. It also provides theoretical insights into the mechanisms of swarm intelligence and their applicability to multi-agent systems, shedding light on the potential of decentralized coordination mechanisms in addressing complex real-world challenges.

1.9 Roadmap

The structure of the remainder of the thesis is as follows. Chapter 2 presents background material and related research in the field, introducing measures of road traffic congestion, adaptive traffic signal control, connected and autonomous vehicles, and self-organization in nature, including concepts of swarm intelligence and Ant Colony Optimization (ACO), as well as RL techniques with a focus on MARL. It also discusses evaluation criteria for selecting algorithms. In Chapter 3, the state of the art is explored, justifying the choice

of MARL over game theory in traffic management, reviewing existing literature on MARL applications in cruise control, collision avoidance, and intersection management, and summarizing any preliminary experiments if any, conducted in the field. Chapter 4 details the methodology, including the design of self-organizing systems for traffic management, the design of pheromone-based communication mechanisms in PERL, and our integration of MARL with pheromone signals for decentralized control in different scenarios. Chapter 5 covers the implementation and simulation environment, providing an overview of the implementation process, software frameworks, and the simulation environment used for testing and evaluation. Chapter 6 focuses on evaluation and results, presenting results from experiments evaluating the proposed methodology, analyzing the performance of the integrated MARL and pheromone-based approach, and discussing implications and areas for improvement. Finally, Chapter 7 concludes the thesis, summarizing key findings and contributions, reflecting on the significance of the research, and suggesting future research directions.

2 Background

This chapter details the concepts and contextual information required to be able to digest the rest of the document conveniently. Some of this information was also present in the confirmation report [13] submitted earlier. It starts with the core problem of congestion and how it is measured, goes on to detail how traffic signal control came about to what it is today, then moves to the rise of disruptive CAV technology which encourages the idea of self-organizing them. The section after that introduces self-organization as a concept and how it is observed in nature and traffic flow. We then discuss communication in swarms as an example of ways to induce self-organization. Finally, we move to the effectiveness of RL as an optimization algorithm and then ponder over the use of MARL as a framework to induce self-organizing behaviour in CAVs as a way to minimize congestion.

2.1 Measures of road traffic congestion

The term flow reminds us of a stream of steady flowing water from one point to another. The flow of water can be measured as the volume of water passing through a point in a given amount of time. If we compared this to traffic flow, it is not too different. Instead of the volume of water, we measure the number of vehicles that pass per unit time. The most predominantly apparent problem caused by poorly optimized traffic flow is the global phenomenon of congestion. Congestion can be characterized by an increase in travel delay and waiting times, higher chances of road incidents and accidents, road rage, inefficient fuel usage leading to excessive costs of maintenance, and carbon emissions harmful to the environment. It is classified into two types, recurring and nonrecurring congestion. Recurring congestion occurs when the demand for the usage of the road exceeds its capacity especially

when there are excessive numbers of vehicles during peak hours, and it forms 45% of the total traffic congestion. On the other hand, nonrecurring congestion is caused by uncertain events like changes in weather, road incidents, construction work, and large sports or music events. It contributes to 55% of the total congestion.

Some ways to measure congestion [25] in road traffic are by the vehicle speeds, the time taken for travel, the delay caused by congestion, and a combination of these. For example, the Speed Reduction Index (SRI) calculates a ratio between the average speed of vehicles in a congested setting and the average speed of vehicles under free-flow conditions.

$$SRI = (1 - \frac{v}{v_{ff}}) \times 10$$

where v is the actual average speed of vehicles and v_{ff} is the average free-flow speed. The ratio is multiplied by a factor of 10 to keep the value between 0 and 10. Only a value less than 4 would indicate non-congested flow. Another similar measure using the speed of vehicles is the Speed Performance Index (SPI), which computes the ratio between the average speed of vehicles to the maximum allowed speed on the road.

$$SPI = \frac{v_{avg}}{v_{max}} \times 100$$

where v_{avg} is the average velocity of vehicles and v_{max} is the maximum possible speed (often the speed limit). The range is shifted between 0 and 100 by multiplying it with a factor of 100. Values less than 25 indicate heavy congestion, less than 50 but more than 25 indicate mild congestion, less than 75 but more than 50 indicate smooth conditions, and any value above 75 indicates very smooth road traffic conditions. Measuring the time taken by a vehicle to cover a specific segment of the road for the actual length of that segment gives us a measure called travel rate, which could alternatively also be calculated from the inverse of the average speed of vehicles.

$$\text{Travel rate} = \frac{T}{L}$$

OR

$$\text{Travel rate} = \frac{1}{v_{avg}}$$

where T is the time taken to cover a road of length L at speed v. Other measures that are based on the travel rate and the time delay caused by congestion are delay rate and delay ratio. Delay rate is simply the difference between the actual travel rate and the acceptable travel rate, which is defined arbitrarily.

$$\text{Delay rate} = \text{Actual travel rate} - \text{Acceptable travel rate}$$

Delay ratio extends this measure to compute the ratio of delay rate to that of the actual travel rate.

$$\text{Delay ratio} = \frac{\text{Delay rate}}{\text{Acceptable travel rate}}$$

A popular measure used by the Highway Capacity Manual [55] is based on the Level of Services approach. From the characterization of congestion itself, we can derive a ratio between the occupancy of a given road segment to the capacity of vehicles it can hold, called the Volume-to-Capacity ratio. This value classifies the congestion into 6 possible classes from A to F, A being free flow condition and F being breakdown of flow.

$$\frac{V}{C} = \frac{N_v}{N_{max}}$$

where V is the occupancy of the road and C is the capacity of vehicles that can fit on the road, N_v is the mean volume while N_{max} is the maximum number of vehicles that a road can hold. More commonly used in urban areas is the Relative Congestion Index (RCI), which is a ratio of the delay in travel time of vehicles and travel time under free-flow conditions. A branch of this measure called the Road Segment Congestion Index utilizes the SPI measure defined previously to measure congestion on a specific road segment. It ranges between 0 and 1, where 0 means heavy congestion.

$$RCI = \frac{(T - T_{ff})}{T_{ff}}$$

where T is the actual time taken for travel, T_{ff} is the time taken for free-flow travel.

Federal congestion measures are those used by the DOT-FWHA including congested hours, travel time index and planning time index. Congested hours are simply the number of hours during specifically identified intervals in a day when the average speed of vehicles in the congested network is less than 90% of the possible speed in free-flow conditions. Similarly, a travel time reliability measure known as the travel time index (TTI) compares the average time taken by vehicles to travel in free-flow conditions with respect to average travel time during the specifically identified peak hours when there are more vehicles in the flow.

$$TTI = \frac{T_{pp}}{T_{ff}} = \frac{v_{ff}}{v_{pp}}$$

where T_{pp} is the peak period travel time, T_{ff} is the free-flow travel time, v_{ff} is the free-flow travel speed,and v_{pp} is the peak period travel speed. The planning time index (PTI) is yet another measure of travel time reliability based on the travel time index. It is a ratio between the 95^{th} percentile travel time to the free-flow travel time.

$$PTI = \frac{T_{95^{th}}}{T_{ff}}$$

where $T_{95^{th}}$ is the 95^{th} percentile of time taken by vehicles for travel and T_{ff} is the average time taken by vehicles in free-flow settings. The percentile quantity here is used to represent one of the worst-case scenarios for travel time. For example, if the PTI value is 2.5, it indicates that vehicles will take 150% more time than the free-flow travel time to reach the destination, and hence should plan to leave that much earlier to be on time for 95% of the trips.

From the above-mentioned measures for congestion, the relevance of TTI and travel rate

to the optimization objective, i.e., minimization of congestion for urban road traffic are valid reasons to choose these as primary metrics to be evaluated.

2.2 Adaptive Traffic Signal Control (ATSC)

The core of Urban Traffic Control (UTC) systems is the actuation of traffic lights at junctions either manually, based on fixed times, or in response to newly acquired data of the traffic flow state [56]. The origin of traffic lights was a simple manually-operated and gas-powered 2-light signal (red and green) indicating to vehicles to stop or go on a particular lane. This was later replaced by electricity-powered traffic lights. The next stage in the evolution of traffic lights was the introduction of fixed-time plans, that had fixed time intervals for the passage of vehicles through a lane. The problem was that these plans could cause delays in light traffic flow conditions due to the unnecessary waiting involved. They also needed to be constantly updated by traffic engineers for it to be of any use, thus also piling additional costs on top of that. A widely known example is the Traffic Network Study Tool (TRANSYT).

Inductive loops installed in the roads at the junction entry points gave rise to vehicle-actuated junctions relying on vehicle presence to change traffic signals. One of the earliest data-driven UTC systems is the microprocessor optimized vehicle actuation (MOVA) that uses data from the inductive loop detector for each lane and controls traffic signals based on its analysis. Systems like MOVA were not considering the effect of decisions made at one junction on the other adjacent junctions. So, it called for the coordinated version of the vehicle-actuated junctions which gave birth to the most commonly used UTC system in the world - Split cycle offset optimisation technique (SCOOT) [41], a centralized approach that detects vehicle presence very frequently making over 10000 optimizations to the split, cycle and offset time for 100 junctions. It also offers flexibility in overriding parameters manually. Other UTC approaches that are in common use include the Sydney Coordinated Adaptive Traffic System (SCATS) [57], Urban traffic optimisation by integrated automation (UTOPIA), Real-time hierarchical optimised distributed and effective system (RHODES), and method for the optimisation of traffic signals in online controlled networks (MOTION).

SCATS is the next most common UTC system in the world. It combines the use of fixed-time plans with data collected from inductive loops built into the pavement at intersections to sense the passage of vehicles and collect information like speed and spacing between vehicles (calculated from the timestamps between two successive detections of vehicles) to dynamically update signal cycle and phase split timings. It primarily operates with many distributed controllers but also has a manual override to switch to centralized control for local junctions. The optimization algorithm responsible for dynamic signal time changes is weak at optimizing the offset time at individual intersections [56] which was needed to manage the effect of change in signal timings on the offsets for the neighbouring junctions.

Since then, more innovative methods of UTC have been researched. [58] used an algorithm inspired by the Oldest Job First (OJF) job scheduling algorithm for traffic signal control using data from vehicular ad hoc networks (VANETs) in connected vehicles. It was done for a single or isolated intersection with no consideration for the effect on other junctions in the network.

2.3 Connected and Autonomous Vehicles (CAVs)

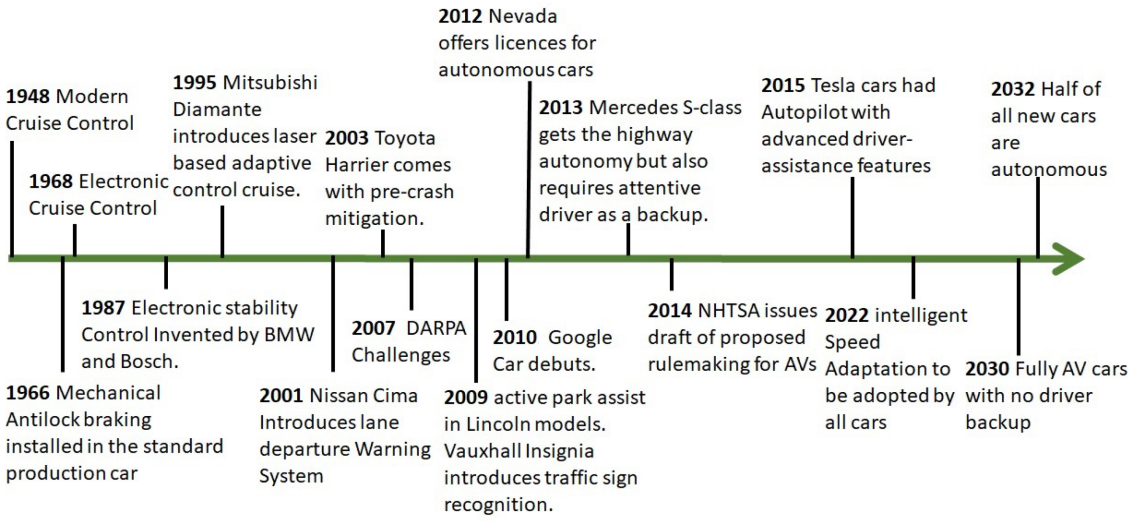

Figure 2.1: Future of CAVs [3]

The emergence of CAVs is a combination of two ground-breaking advancements; one in

autonomous vehicles (AV) technology, which turns vehicles into intelligent agents capable of acting on the environment on their own and the other in connected vehicle (CV) technology, that allows transmission of real-time data between these intelligent entities. This combination favours the idea of decentralized vehicle control by offering highly precise vehicle state data which was not available before and the means to act on that data. It is clear from the forecasted predictions for the coming years in Figure 2.1 that CAV technology is here to stay.

To convert an ordinary vehicle to an autonomous vehicle, it is fitted with a few additional modules that allow it to perceive and act. The perception module used for sensing the world around them can consist of sensors like Radio Detection and Ranging (RADAR), Light Detection and Ranging (LiDAR) and camera.

Vehicular Ad-Hoc Networks (VANETs) [59, 60] is the technology that encapsulates Vehicle-to-Vehicle (V2V) communication, Vehicle-to-Infrastructure (V2I) communication and Vehicle-to-Everything (V2X) communication used in CAVs. Inter-vehicle or V2V communication enables the exchange of information between the vehicles. They could be single-hop for short-range communication or multi-hop systems for long-range communication. Roadside-to-vehicle communication or V2I enables the interaction of vehicles with Roadside Units (RSUs). RSUs can be installed in traffic lights, parking meters, traffic cameras or lane markers. An example of this interaction is the dynamic setting of the speed limit by the RSU. V2X communication involves interaction with relevant components that are not included in either V2V or V2I. For example, V2X powers communication for entities such as pedestrians, by warning them about potential collisions on their smartphones.

With advancements in wireless technologies, conventional vehicles were enhanced with VANET technologies, which made data exchange between vehicles, with the infrastructure and even pedestrians possible. The short-range communication technologies for VANETs include Bluetooth, Bluetooth Low Energy (BLE), Ultra-Wideband (UWB) communication, and ZigBee.

Unlike short-range communication, medium-range communication technologies are especially useful for V2V and V2I interactions. These include Dedicated Short-Range Communication (DSRC) or wireless access in vehicular environments (WAVE). The scalability issue

in DSRC communication is tackled in a long-range communication technology released by the 3rd Generation Partnership Project (3GPP) called Cellular-V2X (C-V2X) technology, which is based on the Long-Term Evolution (LTE) standard. It excels in reliability and speed while supporting both short-range and long-range communications between vehicles and RSUs. This was further improved upon on a later release by utilizing the 5G-new radio interface and radio access technology (5G-NR) standard for 5G-NR V2X, which has close to perfect reliability and low latency, advanced security and can be operated in unison with LTE-based C-V2X technology transmissions so vehicles can access both. From the above, 5G-NR V2X communication technology seems to be the most promising in powering CAVs for traffic flow optimization solutions.

2.4 Self-organization

The origin of the term *self-organization* lies in the fields of physics and chemistry. It was used to explain the formation of macroscopic structures (molecules) or arrangements (physical phenomena) from microscopic processes observed in out-of-equilibrium systems [9] attempting to reestablish equilibrium. The term self-organization is made up of two sub-terms - self and organization. The word organization suggests the concept of change in internal structure or arrangement. And in this case, the word self is referring to the exclusive influence of internal factors or local interactions between constituent parts of a system.

The ingredients required to call a process self-organizing [9] are as follows:

1. Positive feedback: The positive feedback can be thought of as a signal that communicates to the entities that a certain action or state leading from the action is beneficial.
2. Negative feedback: Conversely, negative feedback is like a signal that communicates undesirable states or actions that lead to undesirable states, like a penalty of sorts. Both kinds of feedback are important as they will determine what kind of behaviour or structural arrangement may emerge in the form of stable equilibrium. An everyday occurrence of this kind of feedback can be seen in how parents instil morals in their young person's malleable mindset.

3. Amplification of fluctuations: The randomness in a system is just as essential an ingredient of self-organization since it helps explore new states of the system and new opportunities to discover better arrangements or possible equilibrium states. A simple example would be to imagine 3 people trying to walk through a small opening at the same time. They fail constantly because there are no changes to the arrangement and the same result is obtained as many times as they try the same pattern. The amplification of positive and negative feedback signals for fluctuations assists in bringing the system closer to equilibrium from a randomly discovered starting state that acts as a seed.
4. Multiple interactions: A minimal density of entities to enable them to use not only their own activities but also other's activities as experience. A loose example of this would be study groups, where deriving experience, both things to do, and not to do, from your peers who may have already gone through the same experience before.

From the above-mentioned ingredients, we can say that self-organization can be defined as a âset of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level componentsâ [9]. Self-organization can be characterized more accurately using the following properties [61]:

1. **Autonomy**: Autonomy is the ability of a local entity to perform or act by itself with the complete lack of external control on the system or any of its entities.
2. **Global order**: The global order describes the arrangement or structure that is exhibited by the system as a consequence of the local internal interactions.
3. **Emergence**: Emergence is a phenomenon where the global properties and behaviour that is observed in a system cannot be deduced just from looking more closely at the properties and behaviour of individual components of the system. Emergent behaviour is a consequence of the interaction between local entities of the system. This behaviour is sometimes also simply known âas a whole that is more than its partsâ [62]. Self-organization does not always lead to an emergent phenomenon, but their coexistence in a complex system is quite common.
4. **Dissipation**: The dynamic nature of self-organizing systems with so many moving parts stabilizes at some point to some equilibrium state. This is because there is a loss

of energy in the system after continuously changing states initially. It is analogous to how hot water slowly cools down as the energy that agitates the molecules starts to dissipate into the surroundings.

5. **Instability**: The sensitivity of a self-organizing system to even the slightest of changes in initial conditions and environmental factors can cause large modifications in overall system behaviour. This implies that studying only individual behaviour will not be able to explain the overall system properties that emerge.
6. **Multiple equilibria**: There is more than one possible state where the system is stable, and the structure hardly changes. These states are also called "attractors".
7. **Redundancy of components**: Having a few replacements for the same entity and its behaviour and function in the overall system, makes the entire system robust to local failures of some individual entities. It is like having multiple backups of the same data so losing one data point does not mean total data loss.
8. **Adaptation**: Adapting to any sudden variations in the environmental factors and re-organizing the system to better tackle the situation is an essential property of a self-organizing system.
9. **Complexity**: The complexity of the system comes from the fact that the global properties and behaviour of the system cannot be reduced to a combination of its local constituent properties.
10. **Simple component interaction rules**: The interaction rules between local entities of the system are simple yet cannot be mapped to the global behaviour that it causes, since it was not implied by the description of the interaction rules, i.e., the global interactions are not explicitly clear from the local interaction rules.
11. **Hierarchical pattern**: Two noticeable hierarchies that are formed in a self-organizing system is the level that describes internal interactions or local behaviour, and the level which describes the emergent properties and global behaviour of the system.

Therefore, **self-organization can be defined as a mechanism that allows a system to move towards one of its possible stable states, i.e., reaching a state of equilibria, by continuous rearrangement or reordering caused by interactions between its constituent parts without any explicit external influence**.

A further classification of self-organizing systems [62] reveals that there are two types. Weak self-organizing systems are those in which the re-organization of local components is done by a centralized internal controller. This is sometimes seen in termite colonies where the queen may act as the internal centralized controller. Strong self-organizing systems, in which there is no explicit internal or external central control, are the ones that are usually referred to whenever the term self-organizing system is mentioned. Ant colonies that go out foraging for food form a path to any source of food that is found, and this is a valid example of a strong self-organization system.

Self-organizing systems require the local entities to share information and interact with each other. The mechanisms employed by these systems can be classified using 3 simple factors. Firstly, the type of information that is exchanged between the local entities could either be direct and explicit (marker information) or indirect and implicit (sematectonic information). Secondly, if we consider the flow of information, the relevant information could either be stored in a common information exchange area where all entities come to collect it or the information could be diffused in the environment, analogous to transmission of a broadcast signal. The last factor that can help in this classification is the way the information received by the entities is used. It can be either used to trigger (trigger-based or event-driven) the local entity that received the information or instruction to perform a specific action or to imply (follow-through) multiple ordered actions to be performed by the receiving entity.

2.4.1 In nature

Self-organization is a phenomenon prominent in nature and the real world. The most common examples of self-organization found in nature [61] are shown below.

There are many examples of self-organization in nature (see Figure 2.2). Based on indirect interactions (stigmergy), the **food foraging in ants** is a prime example of self-organization. They leave pheromones on paths and other ants follow this pheromone signal based on a probability that increases with how strong the concentration of the pheromone is. This reinforces the path to the food until the source is depleted. This mechanism was termed 'Swarm intelligence'. The **nest building in termites** is also based on indirect

interactions, where termites use pheromone signals to interact with other termites and increase the chances of placing a brick (which is made of small objects and trash) on top of one that already exists to form columns. The columns that are close to each other pull the termites towards each other due to the proximity of the pheromones and serves to creates an arching shape of the structure, which turns into a dome. Some simple **plants and bacteria use gradient fields** of the pheromone they produce to form clusters (a single supercell) and look for scarce resources in the environment. Gradient fields are based on the concentration of pheromones and the stronger it is, the more individuals are attracted to the point of the supercluster where the concentration is the strongest. This process is formally called **Molding**. Most cells in biology undergo a process called **Morphogenesis**, where the use of the gradient field of a substance called morphogen comes into play. During this process, another cell can tell how far it is from the source of a morphogen by its gradient concentration and can determine its own position (head, abdominal region) in the developing embryo.

Figure 2.2: Self-organization in ants from [4]

The way **spiders beautifully weave webs** is actually because of a very simple process of sematectonic interactions. Individual spiders prefer to walk on draglines (web fastened between two points) instead of the ground, so they have a better chance of fastening the web at the end of another dragline. There is also a small chance for them to move to a

different node and create a new dragline. The balance between these two actions ensures that a web is built. If the probability of walking on a dragline is too high, then the spider is just walking along a dragline without really weaving a web-like structure. If itâs too low, then there will be a lot of nodes and segments scattered without any real connection, so no web is built.

Another type of mechanism seen in ants is based on sematectonic interaction. It ensures that similar objects are placed on the same pile, for example, to sort eggs. Individual ants walk randomly at first and collect objects at random, but place them in bunches where the other objects look like the ones they want to place. This process is called **Brood sorting**. The **flocking of birds, schooling of fish and herding of livestock** are three common phenomena seen in nature all governed by the same 3 simple interaction rules, namely, to maintain a certain distance, a common speed, and centre. In each of these groups, there is at least one sentinel entity that is slightly more sensitive to environmental factors including to the approach of predators. This means they react faster than the other individuals. So when the sentinel entity moves, the other entities follow due to the interaction rules. It is based on sematectonic information. In yet another type of mechanism as seen in lightning bugs, there is a tendency of the individuals to copy the behaviour of the majority of the individuals in the group. It is formally called the Quorum Mechanism. When a minimum (critical) number of entities behave similarly, this phenomenon occurs when the **lightning bugs in a system all start to glow using their internal luminescence**.

2.4.2 In traffic

Real-world problems in various fields like economics, ecology, and even management of state-funded infrastructure (including urban road infrastructure), have a lot of features in common. They are all large scale and have several autonomous entities in the system. The relations between entities may change as and when they leave and reenter the system at any point in time, i.e., the environment is open. The environmental dynamics and that of its components are not predictable due to each local entity holding virtually no or extremely limited knowledge of each other and the system. In all the above-mentioned fields, the entities could be mobile. This means that they are constantly moving, and their

communication range varies continuously. These similarities between different problem areas imply that the challenges that arise from them will also be similar; challenges that are well-suited to be tackled by self-organizing mechanisms.

The self-organizing phenomenon is prominent in multi-agent systems (MASs) since a lot of the requirements for self-organizing systems are satisfied by them. âAn agent is an encapsulated computational system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectivesâ [63]. In a MAS, there are multiple constituent entities or agents that are autonomously acting upon the perceived environment to try and achieve either individualistic goals or one common objective. They can interact with other agents and have a means of communicating and coordinating certain types of behaviour as a whole. Hence, MAS offers a framework for self-organizing mechanisms to be built into software. So, combining the multi-agent system paradigm with self-organizing systems is a reasonable direction to pursue with the hope of solving real-world problems.

For example, [64] use Ant Colony Optimization [65] (ACO) for traffic flow optimization on a simulation of a real-world network in which the traffic model is not dynamic, which means the flow rates are fixed and do not change with time. ACO draws inspiration from ant colonies where the ants rely on pheromone concentrations to signal checkpoints and self-organize a path of ants towards the food source (discussed further in section **??**). [66] tackle a dynamic traffic model with vehicles whose destinations are tagged by colours and the concept of stench pheromones which prevents every vehicle from choosing the same best route and causing congestion. The control of vehicles is based on Model Predictive Control and raises scalability concerns. Similarly, [67] add in the coordination aspect with some additional rules depending on the size of the platoon and availability of road after the intersection in order to maximize the efficiency of green light phases. The self-organization of traffic lights is supported by a dynamic coordination mechanism that allows for green waves across multiple adjacent intersections at close proximity. In [68], it was implied that self-organizing mechanisms are well-suited for decentralized completion of tasks by agents and for the formation of clusters or groups of agents that have a common task or goal. Similar to molding in nature discussed previously, one of the emergent behaviours

in multi-agent self-organizing traffic systems is the platooning of vehicles [48], which is when clusters of vehicles form groups and maintain lateral and longitudinal speed. Detector data passed in communication between neighbouring intersections in [69] helps prediction of arrival times of vehicles to the next intersection which in turn, makes it easier to find the optimal signal timing. Another example of self-organization used for Urban Traffic Control (UTC) [70] is derived from gradient fields of pheromones used in nature [71]. The concept of co-fields, or computational fields is analogous to how objects move towards other objects based on gravity fields. Self-organizing multi-agent systems are scalable and robust to failures of individual components. They learn to adapt to the changing environment and local interactions. However, a large chunk of current research in self-organizing multi-agent systems was quite theoretical [72].

The scalability of the above examples can be called into question when considering real-world traffic situations. The inter-vehicle communication technology powering CAVs now encourages proposed research to go a step further for distributed vehicle control supported by information exchange that can potentially induce self-organization in individual vehicles for traffic flow optimization.

2.5 Swarm intelligence

The problem of coordinating multiple self-organizing entities using a decentralized manner of control in a system gave rise to a group of methods capable of guiding the entities to attain a global objective. These were called Swarm Intelligence (SI) methods [73]. Two well-known methods that draw inspiration from natural systems are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

2.5.1 Ant Colony Optimization (ACO)

In ant colonies, the concentration of pheromones left by fellow ants causes ants to follow and reinforce the shortest path towards a food source. ACO [65, 74] works on the same basic principle of stigmergy, an indirect method of communication by making local modifications to the environment only perceivable by other individuals in the same local area. To explain

the working of ACO, the task chosen is distance optimization of the path to the food source.

For example, if the given path has only a single decision point between two different routes, then given a choice between two different paths, in the presence of no pheromone signals in either of the paths, i.e., no other ant has ever traversed through either of the other paths, the ant is forced to choose randomly between the two. This means that the first ant to traverse through the scenario will randomly pick between the two possible paths. Out of these, the path chosen gets pheromone deposited to attract other ants. In the case of differently sized paths, the pheromone deposited on the shorter path by stochastic fluctuations in behaviour (Figure 2.3), due to its shorter length, has larger concentrations of pheromone deposited earlier in the process since the ants return to the nest faster, and the other ants are encouraged to traverse it more often than the longer path. Ant System (AS) is the earliest example of an ACO algorithm. Improvements focused on the update of the pheromone signal over the AS algorithm have lead to better algorithms like MAX-MIN Ant System (MMAS) and Ant Colony System (ACS). Though all ACO algorithms work on the basis of the same principle, there are minor differences between them as examined in Table 2.1.

Algorithm	**Pheromone Update Equation**
AS	$(1-\rho)\cdot\tau_{ij}+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$
MMAS	$[(1-\rho)\cdot\tau_{ij}+\Delta\tau_{ij}^{best}]_{\tau_{\min}}^{\tau_{\max}}$
ACS	$\begin{cases}(1-\rho)\cdot\tau_{ij}+\rho\cdot\Delta\rho_{ij} & \text{if } (i,j) \text{ is the best}\\ \tau_{ij} & \text{otherwise}\end{cases}$

Table 2.1: Differences in pheromone update equations for ACO algorithms

τ_{ij} is the pheromone concentration on the edge connecting points i and j, ρ is the evaporation rate of the pheromone that mimics the same behaviour as observed in natural ant pheromones, m is the number of ants, and the amount of pheromone deposited by the ant k can be calculated based on the inverse of the length of the route. This means that the smaller the route, the larger the concentration of pheromone deposited. MMAS improves upon AS by making pheromone updates only on the knowledge of the best ant ($\Delta\tau_{ij}^{best}$), and by bounding the pheromone update using a minimum and maximum value that are tuned based on the problem. ACS goes further beyond and improves upon the algorithm by introducing a local pheromone update which helps to expand the search for possible paths

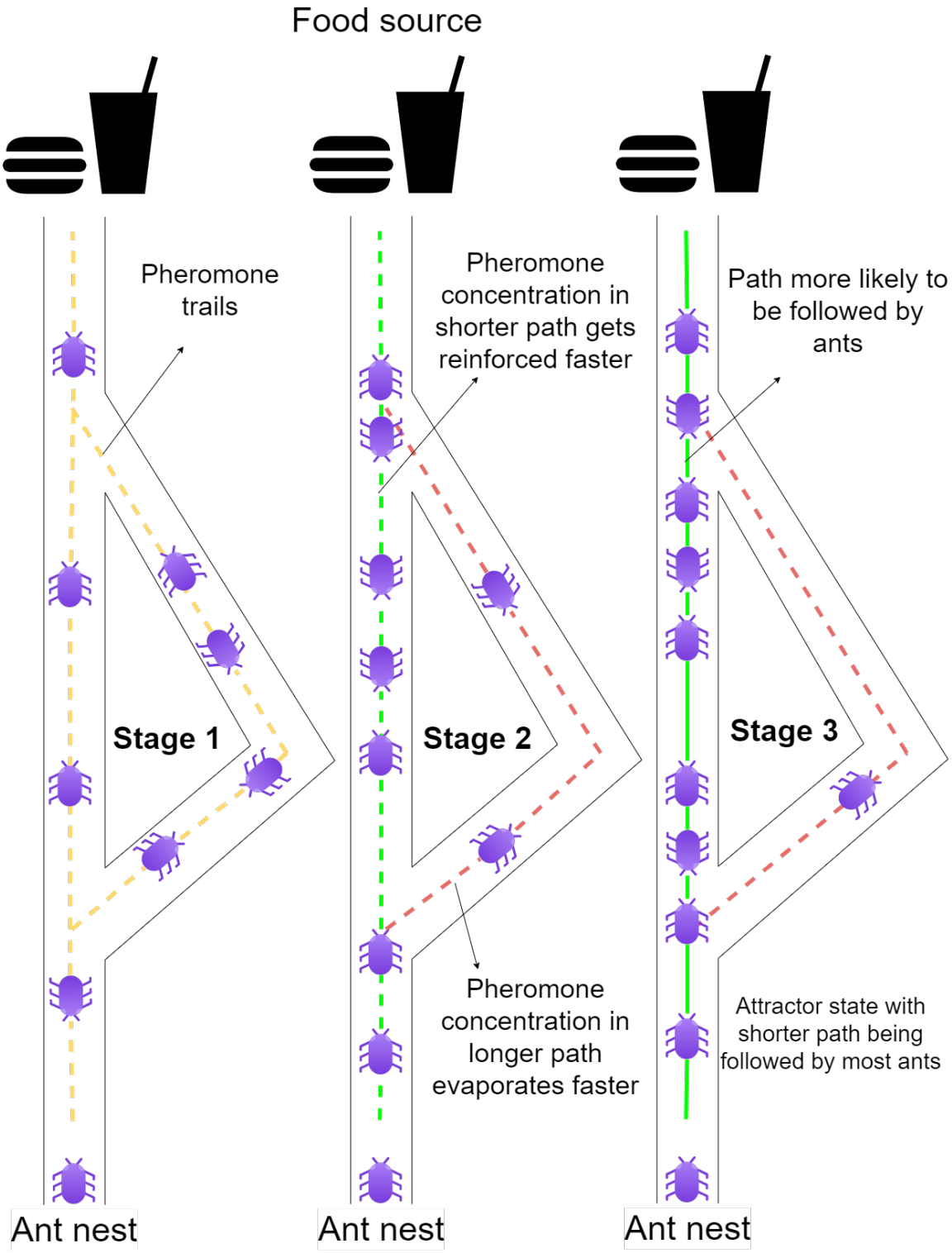

Figure 2.3: ACO process in Ants

and encourage individual entities to explore during an iteration by reducing the probability of individuals to exploit what another individual may have already explored. This is done using the pheromone decay coefficient (φ) which updates the pheromone values offline as follows:

$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0$

where τ_0 is the initial value of the pheromone concentration and $\varphi \in (0, 1]$

Particle Swarm Optimization (PSO)

According to the sociobiologist in [75], the advantages gained from individuals utilizing experiences/discoveries of other individuals in the group outweighs the disadvantages of induced competition between them for unpredictable distributions of resources. The core of PSO [76] lies on the principle that each particle (individual) of the swarm (group) holds a possible solution to the optimization problem. The particles then use globally available information to iteratively update their individual behavioural properties until the desired behaviour or goal is attained. PSO was inspired by the natural system observed in the process of flocking of birds. It started as a simple simulation of the social environment in which these animals interact to locate a food source. The movement was constrained using a rule that matched the velocity of the closest neighbouring bird or agent. The concept of craziness, analogous to the stochastic fluctuations in self-organizing systems was introduced to allow a more natural rate of synchronization of the movement of the flock instead of the quick and abrupt unanimous movement caused by hardcoded rules. As the concept of a roost or an objective point was introduced, it was seen that the craziness variable became obsolete since the roost location was a sufficient motivating factor for individual birds to choose their best velocities. The roost location symbolized a food source like a cornfield. Vectors (cornfield vectors) called pbestx and pbesty held a list of the x and y coordinates that were thought to be the best according to each agent in the system. An additional variable called gbest that held the index of the agent that had the best coordinates compared to all other agents with respect to the cornfield was also added. Though previously the updates to velocities were made using only the inequalities of the coordinates, it was realized that taking the difference between the present and best coordinates yielded better results. Both

the craziness factor and the velocity matching rules were dropped since the search seemed better off without them.

The simplified version of PSO for a multi-dimensional problem looks like

$$vx[][] = vx[][]+2*rand()*(pbestx[][]-presentx[][])+2*rand()*(pbestx[][gbest]-presentx[][])$$

For example, if a particle's velocity behaviour is to be updated on the x-axis, the equation above shows that it uses a combination of global knowledge from the best-known x-coordinate globally ($pbestx[][gbest]$) and the best-known x-coordinate locally by the agent ($pbestx[][]$). The double square brackets are to indicate a multi-dimensional vector. A SI approach is possible in situations where certain principles are followed. Firstly, the swarm should be able to carry out simple computations (proximity). Secondly, they should be responsive to factors in the environment (quality). Thirdly, the response of each particle in the swarm should not be nearly the same and should display some level of diversity. Lastly, the behaviour of the swarm should change only when it steps closer to the global objective.

Like ACO algorithms, PSO is also an iterative algorithm. A qualitative analysis [73] of the two has proven the superiority of the ACO algorithm in the distance optimization scenario so frequently referenced.

2.6 Reinforcement Learning (RL)

The term 'Reinforcement Learning' (RL) has many definitions and explanations, as perceived by people from different areas of the scientific community. The intelligent entities capable of perceiving and acting on their surroundings are called agents. The surroundings that the agents act on is the environment. The virtual representation of the environment as perceived by the agent is the state or observation. The next state that the agent would end up in, along with the reward signal for the action taken or state the agent is in is provided to the agent by the environment. **The process of an agent learning in an unknown environment by trial-and-error of actions making use of the reward signals provided to formulate a plan that can help make better decisions in time is called reinforcement learning**. For example, even a baby learning to walk can be modelled as a reinforcement learning process.

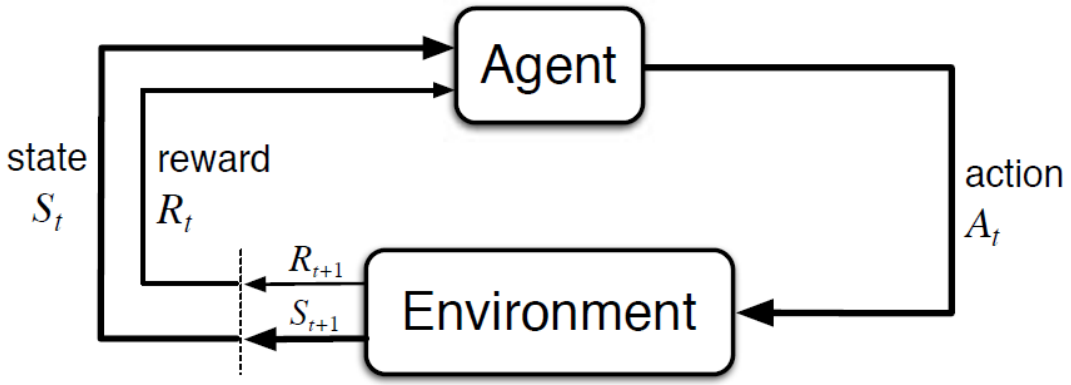

Figure 2.4: MDP for RL as described in [5]

Formally, an environment in RL can be better visualized as a Markov Decision Process (MDP). An MDP is a framework for defining a problem that can be solved using RL. It is based on the Markov property which enunciates that the next state is only dependent on the current state of the environment, and all the states leading up to it are irrelevant. A visual representation of the process can be examined in Figure 2.4. "The MDP framework is a considerable abstraction of the problem of goal-directed learning from interaction" [5].

The behaviour of the agent in the environment is controlled by a policy or a plan that tells the agent how to act in any given situation of the environment. It will always try to improve its policy to collect more rewards and eventually maximize them. For the agent to learn and start choosing better actions, it needs to update its policy. While using the immediate reward signals as the basis for these updates sounds like a good idea, it works even better when the policy is tuned using an estimate of what the total reward collected from that state would be, assuming it follows a certain policy. This is what the value function does as represented in equation 2.1.

$$v_\pi(s) = E_\pi[G_t|S_t = s], \forall s \in S \tag{2.1}$$

where $v_\pi(s)$ denotes the predicted total value that an agent will collect in rewards from state s onwards if it follows the policy π defined by

$$\pi(a|s) = P[A_t = a|S_t = s], \forall a \in A(s), s \in S \tag{2.2}$$

where $P[A_t = a|S_t = s]$ is the probability distribution of actions given the state.

The expected return G_t in equation 2.1 can be computed as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^k R_{t+k+1} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.3}$$

in which γ is called the discount factor which can vary from a scale of 0 to 1 depending on how much importance the agent wants to give to future rewards for value calculation, and R_t represents the immediate reward at time t. The discount factor is inspired by animal behaviour that shows a higher preference for immediate rewards.

An optional constituent in RL is the model that predicts the dynamics of how the environment would react to the actions of the agents, in terms of what state will be observed by the agent when it performs a certain action. This is called the state transition probability function. The presence or absence of this component creates a split between two broad classifications in RL methods; model-based and model-free methods. Model-based RL algorithms can make use of the transition probability function and make it sample-efficient. However, model-free algorithms avoid the complexities of learning a model and instead learn just by interacting with the environment. It is quite difficult to generate a perfect model of the environment in real-world problems, and as such model-free algorithms are more popular.

A policy that can select actions such that it maximizes the expected return is the optimal policy. "There is always at least one optimal policy that is better than or equal to the other policies" [5]. There could also be more than one, and they all will have the same value function. The optimal value function is obtained using the recursive Bellman-optimality equation:

$$V_*(S) = \max_a [R_s^a + \gamma \sum_{S'} P_{SS'}^a V_*(S')] \tag{2.4}$$

where $V_*(S)$ is the optimal value for a given state S observed by the agent, R_s^a is the immediate reward obtained by the agent upon taking an action a in state S, γ is the discount factor, $P_{SS'}^a$ is the probability transition function for state S to state S' by taking an action a, and $V_*(S')$ is the optimal value function for the next state S'. A policy

$\pi \geq \pi' \iff v_\pi(s) \geq v_{\pi'}(s) \forall s \in S$. This means that the optimal policy π_* is the one who's value function is $v_*(s) = \max_\pi v_\pi(s)$ (optimal).

RL algorithms are sometimes also classified on the mechanism by which the actions are picked. In policy-based methods, an explicit policy that maps a state to action is built and stored in memory. Conversely, in value-based methods, a value function is stored and the policy is derived by choosing the actions greedily based on the value. Though storage of these values and policies was done using a tabular-based method earlier, most algorithms have moved to use function approximation owing to its cheap computation and memory resource usage even with increasing state and action dimensions. This function approximation is often done using neural networks and leads into the area of Deep Reinforcement Learning (DRL). Algorithms that combine the strengths of the value-based and policy-based mechanisms are called actor-critic methods, where the actor-network represents the approximation of the policy and the critic-network represents the value function predictions. REINFORCE [5] is a policy gradient algorithm, while Deep Q-Networks (DQN) [77], Double DQN [78] and Dueling DQN [79] are common examples of value-based methods showcased in multiple Atari games to beat human-level performance. Some notable examples of astonishing results achieved using actor-critic methods are displayed in simulated experiments involving robot locomotion, walking, and even navigating 3D mazes from visual input using Proximal Policy Optimization (PPO) [80], Trust-Region Policy Optimization (TRPO) [81] and Asynchronous Advantage Actor-Critic (A3C) [82]. All the above-mentioned algorithms are model-free. The most notable examples of model-based algorithms are AlphaZero [83] for chess gameplay good enough to beat a world-champion, AlphaStar's [84] multi-agent RL algorithm that reached the rank of grandmaster (among the top 0.2% of human players) in a real-time strategy game called StarCraft II, and the recent groundbreaking AlphaFold [85] for protein folding with biological applications including medicine. RL has been a successful approach to many real problems, even UTC.

Deep Q-Networks (DQN) [28], a reinforcement learning algorithm was used by [21] to reduce travel time at an urban intersection. Similar research was done in [29] where the traffic state is detected using a convolutional neural network (CNN) [86] that then merges into the DQN layers. However, all the above-mentioned work was done on a single or isolated

intersections and do not consider the effects of signal changes in the said intersection on other intersections in the network, i.e., it lacked coordination between multiple intersections. [35] uses a centralized DQN controller for multiple (4 intersections with 4 possible phases = 4^4 = 256 possible actions) intersection signal control. It uses a CNN on the images of the traffic states at said intersections. The CNN extracts relevant features and feeds them into the DQN as input. The centralized controller, though not requiring any explicit communication for coordination, will not scale as the number of intersections increases. Another ATSC method [31] uses the YOLO (You Only Look Once) [87] object detection algorithm to generate traffic density state information from a static image, which is fed into the Q-Learning algorithm as input to select the green phase timing for the intersection. Similarly, [33] used Actor-Critic methods (A2C and A3C respectively) [82] to select phase timings for traffic lights in order to reduce waiting times and delay of vehicles at the intersection. All the work done so far is either focused on isolated intersections without any coordination with adjacent intersections or use a centralized approach for control, which won't scale with the number of CAVs as agents in the real world. This is why a framework that integrates the learning techniques of RL into systems with multiple entities is required.

2.6.1 Offline RL

In Offline RL, the collection of data/experiences and the learning happen on different threads. The focus lies on leveraging past experiences to inform decision-making processes. This stands in contrast to traditional RL techniques that rely on continuous interaction with the environment to learn and refine strategies. The challenge with offline RL is ensuring that the policy derived from past experiences remains applicable and effective in novel situations.

2.6.2 Online RL

In Online RL, both the collection of experiences and the learning happen simultaneously. The primary focus shifts towards learning and refining decision-making strategies through continuous interaction with the environment. Unlike Offline RL, which relies on pre-existing datasets, Online RL algorithms actively engage with the environment in real-time to gather

information and adapt their policies accordingly. The challenge lies in striking the right balance between exploring new possibilities and exploiting known strategies to maximize long-term rewards.

Features	**Online RL**	**Offline RL**
Speed	Slow (✗)	Fast (✓)
Memory Usage	Low (✓)	High (✗)
Learning Efficiency	High (✓)	Low (✗)
Sample Efficiency	Low (✗)	High (✓)
Real-time Feedback	Yes (✓)	No (✗)
Batch Processing	No (✗)	Yes (✓)
Exploration	Continuous (✓)	Fixed (✗)
Scalability	Limited (✗)	High (✓)
Data Collection	Simultaneous (✓)	Pre-collected (✗)
Suitability for MARL	Low (✗)	High (✓)

Table 2.2: Comparison of Online and Offline RL for Self-Organization in MARL

2.6.3 Model-free RL

In Model-free RL, the learning process occurs without explicit knowledge or modeling of the underlying environment dynamics. Instead, agents directly learn optimal policies through trial-and-error interactions with the environment. This approach is particularly suited for scenarios where the environment is complex or poorly understood, making it challenging to construct an accurate model. Model-free RL algorithms, such as Q-learning and SARSA, focus on iteratively improving policies based solely on observed experiences without the need for a predefined model. However, one of the main challenges with model-free RL is the potential for inefficient exploration and slow convergence, especially in high-dimensional or complex environments. Additionally, model-free RL may struggle to generalize well to unseen situations, limiting its applicability in certain real-world settings. This is also currently the more popular type of RL today.

2.6.4 Model-based RL

In contrast to Model-free RL, Model-based RL involves explicitly modeling the dynamics of the environment. Agents use these learned models to simulate possible future scenarios and plan actions accordingly. By incorporating a model of the environment, Model-based RL

algorithms can potentially achieve more efficient and effective decision-making compared to their model-free counterparts. However, constructing accurate models of complex environments can be challenging, especially in scenarios with high uncertainty or non-linear dynamics. This also means that model-based algorithms may suffer from computational complexity and require significant computational resources for model learning and planning. Despite these challenges, model-based RL offers the advantage of improved sample efficiency and the ability to make informed decisions even in situations where direct exploration is costly or impractical.

Features	**Model-Free RL**	**Model-Based RL**
Flexibility	High (✓)	Low (✗)
Sample Efficiency	Low (✗)	High (✓)
Computational Efficiency	High (✓)	Low (✗)
Planning Horizon	Short-term (✗)	Long-term (✓)
Accuracy	Not always accurate (✗)	More accurate (✓)
Robustness	Sensitive to noise (✗)	More robust (✓)
Realistic	Closer to real-world problems (✓)	Not realistic (✗)
Learning Stability	May suffer from instability (✗)	More stable learning (✓)

Table 2.3: Comparison of Model-Based and Model-Free RL

2.7 Evaluation criteria used to select algorithms

2.7.1 Learning efficiency

Learning efficiency measures how rapidly an agent can learn optimal policies or value functions from interactions with the environment. A highly learning-efficient algorithm can effectively utilize available data to make informed decisions and adapt its strategies in response to changes in the environment. Evaluating learning efficiency involves assessing factors such as convergence speed, computational complexity, and the ability to generalize learning across different scenarios.

2.7.2 Sample efficiency

Sample efficiency reflects how well an agent can derive meaningful insights and make informed decisions with minimal interaction with the environment. An algorithm with high

sample efficiency requires fewer samples or experiences to achieve comparable performance to algorithms that demand larger datasets. Evaluating sample efficiency involves examining the trade-off between the **quality of learning** and the amount of data required, as well as assessing techniques for effective exploration and exploitation of available information.

2.7.3 Exploration

Exploration refers to the strategy employed by an RL algorithm to discover new states or actions in the environment. Effective exploration is crucial for discovering optimal policies and avoiding premature convergence to suboptimal solutions. Evaluation of exploration strategies involves assessing the algorithm's ability to balance between exploiting known information to maximize immediate rewards and exploring new possibilities to uncover potentially better solutions. Factors such as exploration-exploitation trade-offs, exploration strategies (e.g., epsilon-greedy, optimistic initialization), and the ability to handle exploration in high-dimensional or complex environments are considered.

2.7.4 Scalability

Scalability measures aspects such as computational efficiency, memory usage, and algorithmic complexity. Evaluating scalability involves analyzing how well an algorithm can handle larger datasets, more complex environments, or increased computational demands without sacrificing performance or efficiency. Factors such as parallelization, distributed computing, and algorithmic optimizations are considered.

2.7.5 Realism

Realism evaluates how accurately the algorithm captures real-world environment dynamics and complexities (model-based vs. model-free argument). Evaluating realism involves comparing the algorithm's behavior and performance in simulation or real-world environments to expected real-world outcomes and assessing its ability to generalize beyond training data.

2.7.6 Hyperparameter Sensitivity

Hyperparameter sensitivity assesses how significantly an algorithm's performance is affected by variations in its hyperparameter settings. Some require precise tuning of hyperparameters such as learning rate, exploration-exploitation parameters, or network architectures. Highly sensitive algorithms may suffer from instability or degraded performance if hyperparameters are not optimally configured. In this research, we prioritize algorithms with low hyperparameter sensitivity, as our primary focus is on analyzing the integration of pheromones with MARL to induce self-organization.

2.7.7 Ease of Use

Ease of use considers factors such as ease of implementation, configuration, and integration into existing systems. Evaluating ease of use involves examining documentation, code readability, availability of libraries or frameworks, and support for common programming languages. Additionally, user-friendly interfaces, tutorials, and examples contribute to the overall ease of use of an RL algorithm. Factors such as flexibility, modularity, and compatibility with standard development tools are also considered when evaluating ease of use.

Table 2.4: Comparison of Selected Algorithms Based on Evaluation Criteria

Algorithm	**LE**	**SE**	**EX**	**SC**	**RE**	**EU**	**HS**
Q-Learning	Mod	Low	Basic	Poor	Low	High	High
DQN	High	Mod	Imp.	Mod	Low	Mod	High
PPO	High	High	Adv.	High	Mod	Mod	Mod
TD3	High	High	Soph.	High	Mod	Mod	Mod

Legend: LE = Learning Efficiency, SE = Sample Efficiency, EX = Exploration, SC = Scalability, RE = Realism, EU = Ease of Use, HS = Hyperparameter Sensitivity. Low = Poor, Mod = Moderate, High = Strong, Basic = Simple exploration (e.g., ϵ-greedy), Imp. = Improved (e.g., experience replay), Adv. = Advanced (policy-based), Soph. = Sophisticated (e.g., twin critics).

2.8 Algorithms used

2.8.1 Q-Learning

Q-Learning [88] is a fundamental algorithm that learns the optimal action-value function directly from observed experiences. It iteratively updates estimates of the quality of actions (Q-values) based on the observed rewards and transitions between states. Q-Learning is known for its simplicity and ability to handle discrete action spaces efficiently. However, it may struggle with large or continuous action spaces due to the need to maintain a separate Q-value estimate for each action.

2.8.2 Deep Q-networks (DQNs)

Deep Q-networks (DQNs) [89] represent a significant advancement in RL, specifically in handling high-dimensional state spaces encountered in complex environments. Unlike traditional tabular methods like Q-Learning, which store Q-values for each state-action pair in a table, DQNs leverage the power of deep neural networks to approximate the action-value function.

The key innovation of DQNs lies in their ability to learn a mapping from raw sensory inputs (such as images or sensor readings) directly to Q-values, without the need for explicit feature engineering. By utilizing deep neural networks, DQNs can effectively process complex and continuous state spaces, making them suitable for tasks with high-dimensional observations.

However, training DQNs poses several challenges. One significant issue is the instability and divergence that can occur during training. This instability often stems from the non-linear nature of neural networks and the correlations present in consecutive observations. As the neural network parameters are updated during training, the Q-values may fluctuate, leading to suboptimal or divergent behavior.

To address these challenges, researchers have proposed various techniques and improvements to stabilize DQN training. These include experience replay, where past experiences are stored and sampled uniformly during training to break correlations between consecut-

ive observations. Additionally, techniques such as target networks and double Q-learning have been introduced to mitigate the divergence issues by stabilizing the Q-value estimates during training.

Despite these challenges, DQNs have demonstrated remarkable success in various domains, including video games, robotics, and autonomous driving. Their ability to learn directly from raw sensory inputs and handle high-dimensional state spaces makes them a powerful tool for tackling complex RL problems.

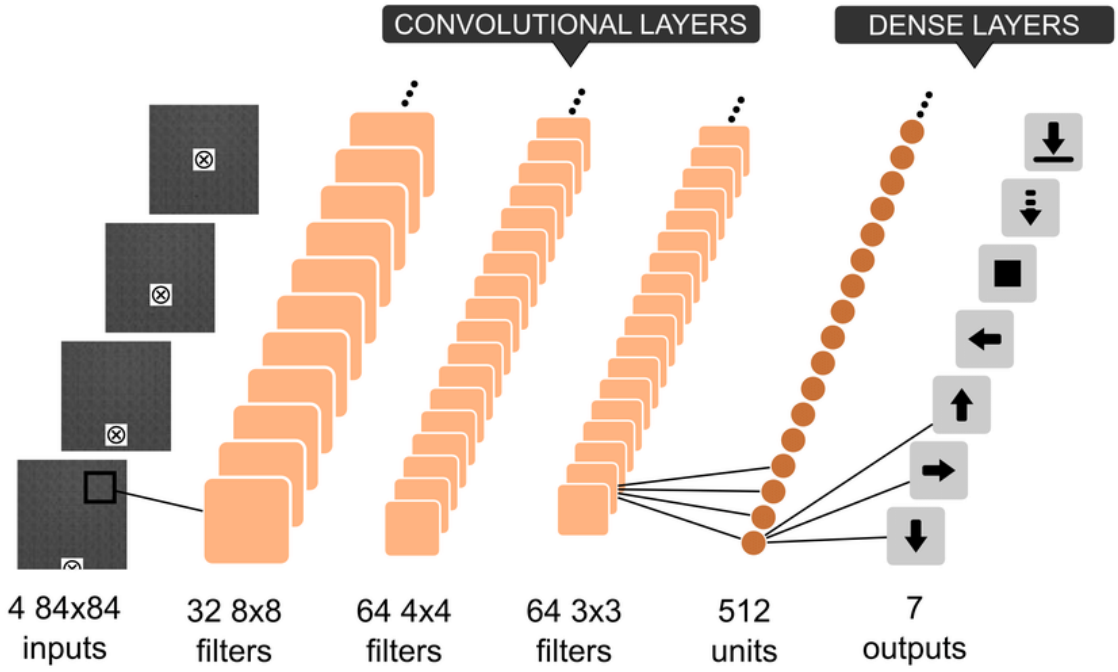

Figure 2.5: DQN architecture [6]

2.8.3 Proximal Policy Optimization (PPO)

Proximal Policy Optimization [90] (PPO) is a policy optimization algorithm that aims to maximize the expected cumulative reward by iteratively updating the policy parameters. Unlike value-based methods such as Q-Learning and DQN, PPO operates directly on the policy function, adjusting the policy parameters to improve performance. PPO is known for its simplicity, stability, and sample efficiency, making it a popular choice for training deep RL agents, especially in environments with continuous action spaces.

The core idea behind PPO is to iteratively update the policy parameters while ensuring that the updates are "proximal" to the current policy. This means that the updated policy

remains close to the current policy, preventing drastic changes that could lead to instability during training. By controlling the magnitude of policy updates, PPO aims to maintain stability and improve sample efficiency. Despite its simplicity, PPO has been shown to achieve competitive performance across a wide range of RL tasks, including continuous control and robotic manipulation.

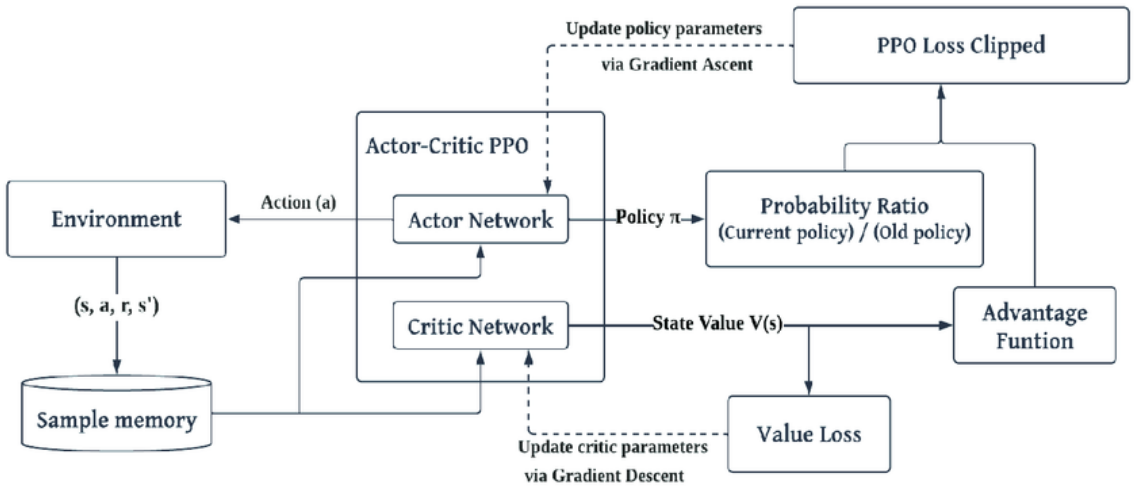

Figure 2.6: PPO architecture [7]

It is known for its stability during training, which is crucial for effectively learning complex policies in high-dimensional state and action spaces. By constraining the magnitude of policy updates and incorporating techniques such as clipped surrogate objectives, PPO mitigates issues such as policy oscillations and divergence, ensuring smooth and efficient learning. Another advantage is its sample efficiency, meaning it can achieve good performance with fewer training samples compared to other RL algorithms. Overall, PPO is a robust and effective algorithm for training RL agents, especially in environments with continuous action spaces.

2.8.4 Twin Delayed DDPG (TD3)

Twin Delayed DDPG [91] (TD3) is an improved version of the Deep Deterministic Policy Gradient [92] (DDPG) algorithm that addresses several limitations and challenges. TD3 introduces twin critics and delayed policy updates to enhance stability and robustness during training. By using two separate critic networks to estimate the Q-value, TD3 reduces the overestimation bias and improves the quality of value estimates. Additionally, the

delayed policy updates in TD3 help stabilize the learning process and prevent premature convergence, leading to more robust and efficient training.

In TD3, the main focus is on learning the Q-function (action-value function), which estimates the expected return for taking a particular action in a given state. The algorithm aims to improve the accuracy and stability of the Q-function estimates by using twin Q-functions and delayed updates. These enhancements help mitigate overestimation bias and improve the performance of the value function approximation.

While TD3 primarily focuses on learning the Q-function, it still utilizes a deterministic policy, similar to DDPG. The policy in TD3 is learned alongside the Q-function, but it is used mainly for exploration purposes rather than being the primary driver of action selection. Therefore, despite having a policy component, TD3 is fundamentally a value-based algorithm.

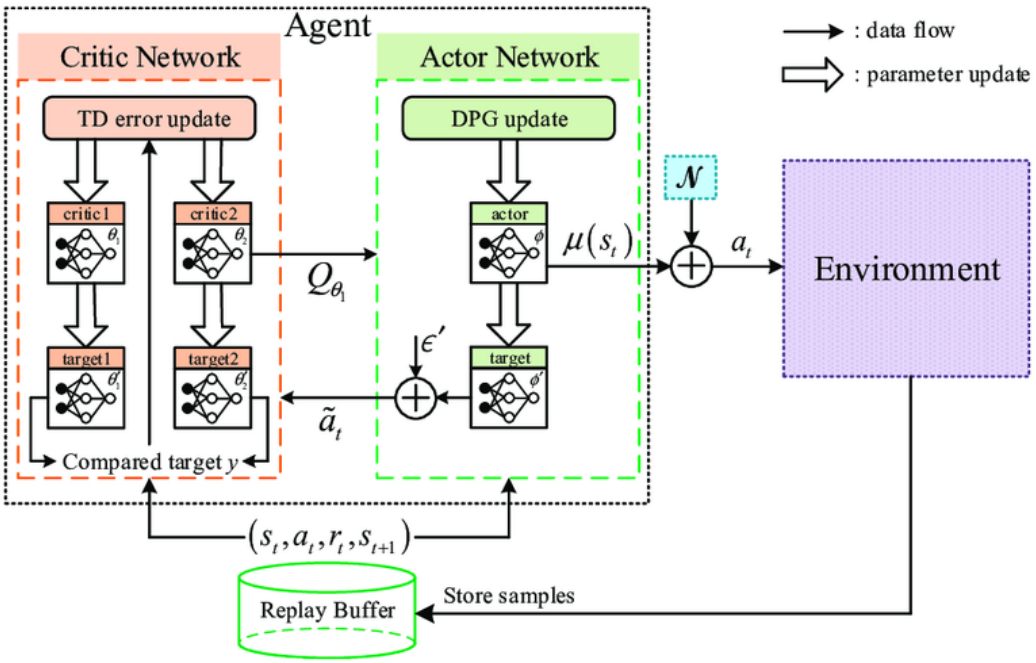

Figure 2.7: TD3 architecture [8]

In summary, TD3 is classified as a value-based reinforcement learning algorithm, but it incorporates elements of policy learning to improve exploration and learning stability.

Based on the evaluation of the listed criteria, **PPO** and **TD3** were selected as the most suitable algorithms for this research.

Algorithm	Offline/Online	Model-based/Model-free	Value-based/Policy-based
Q-learning	Online	Model-free	Value-based
DQN	Online	Model-free	Value-based
PPO	Online	Model-free	**Actor-Critic**
TD3	**Offline**	Model-free	Value-based

Table 2.5: Classification of algorithms used

2.9 Multi-agent Reinforcement Learning

Multi-agent Reinforcement Learning (MARL) is an extension of RL that deals with scenarios where multiple autonomous agents interact with each other and their environment. Unlike single-agent RL, MARL involves coordination, competition, and collaboration among agents, making it applicable to a wider range of real-world problems such as multi-robot systems, autonomous vehicles, and cooperative gaming environments.

Decentralized MARL presents a promising approach for coordinating autonomous agents in complex environments. However, there are several challenges that must be addressed to realize its full potential:

2.9.1 Curse of Dimensionality

The exponential growth of state and action spaces with the number of agents poses significant computational overhead, limiting the scalability of MARL algorithms. This is often addressed through the Centralized Training Decentralized Execution (CTDE) paradigm, which involves training a single centralized policy that is shared among multiple agents. During execution, each agent operates autonomously with its own copy of the policy. This approach helps to mitigate the computational complexity associated with the exponential growth of state and action spaces as the number of agents increases.

2.9.2 Partial observability

Partial observability refers to the scenario where an agent does not have complete information about the environment or the actions of other agents. This lack of complete information can pose challenges in decision-making and learning for the agent. Partial observability complicates the learning process as agents must reason about the hidden information and infer

the intentions and behaviors of other agents based on the partial observations available to them. This need for coordination in partially observable environments is mostly addressed by strategies for communication [14, 15] between the agents or belief state estimation [16].

2.9.3 Non-Stationarity

The actions and policies of other agents continuously change as they each perform their own actions. Adapting to dynamic and evolving environments while maintaining stable and effective policies presents a challenge [17] for decentralized MARL algorithms.

The standard approach for addressing non-stationarity in MARL involves considering information about other agents and reasoning about them. However, existing approaches are not always able to effectively handle sparse good experiences in a scalable manner [18]. Information-theoretic constraints on methods for MARL in mixed cooperative and competitive games have been explored to address non-stationarity in [19].

Some research on the integration of optimal transport theory with MARL has been explored as a means of addressing non-stationarity [93].

[94] explores combat scenarios within MARL where agents adapt to non-stationary environments.

To improve MARL performance in the face of reward sensitivity and non-stationarity, a hybrid framework called RACE has been proposed, utilizing Evolutionary Algorithms (EA) [95]. Overall, addressing non-stationarity in MARL remains a key focus of research, with various approaches and frameworks being developed to tackle this challenge.

2.9.4 Learning from Sparse Rewards

Delayed or infrequent rewards, or feedback only for specific actions, pose challenges for learning effective policies in both single-agent RL and decentralized MARL. Agents may struggle to learn informative gradients or signals for updating their policies, leading to slow convergence and suboptimal performance. A popular way this is addressed is Hindsight experience replay [96], where the key idea is to leverage failed experiences encountered during training by relabeling them to turn them into learning opportunities.

2.9.5 Coordination and Consensus

Achieving coordination and consensus among decentralized agents with conflicting objectives or limited communication capabilities is a fundamental challenge in MARL. Designing mechanisms for resolving conflicts, negotiating agreements, and achieving global convergence poses significant technical and algorithmic challenges.

Current MARL methods primarily focus on improving individual policies, neglecting group-level policies and resulting in weak cooperation. To address this issue, a novel Consensus-oriented Strategy (CoS) has been proposed, emphasizing both group and individual policies simultaneously [97]. Distributed cooperative MARL frameworks typically assume undirected coordination graphs and communication graphs without utilizing any consensus algorithm [98]. These attempts highlight the importance of addressing coordination and consensus in MARL to enhance cooperation among agents.

2.9.6 Emergent Complexity

Decentralized MARL can give rise to the emergence of unintended behaviors or conflicts among agents. Understanding and guidance of this emergent behavior in the system requires coordination mechanisms to ensure stability and effectiveness.

MARL is being used as a key tool in addressing emergent complexity in decentralized systems, particularly in the context of distributed intelligent systems [99]. [100] discuss techniques for centralized training of MARL using DQN as the baseline model, highlighting the importance of communication between agents, as a means of coordination.

Decentralized signal control in multi-modal traffic networks [101] may use direct communication strategies to address emergent complexity. On a similar thread, [102] achieves multi-agent cooperation through unsupervised learning of joint intentions.

There is also research that focuses on explainable Artificial Intelligence (AI) [103] to gain insights into agent behavior in cooperative MARL, highlighting the importance of understanding agent training behavior in addressing the phenomenon of emergence.

The existing indicates a growing interest in addressing emergent complexity in MARL through various approaches and frameworks, emphasizing the importance of communication,

coordination, and understanding agent behavior in multi-agent systems.

2.10 Summary

This chapter provided a comprehensive background on key concepts relevant to this research. It covered measures of road traffic congestion, adaptive traffic signal control (ATSC), and the role of connected and autonomous vehicles (CAVs). The discussion on self-organization and swarm intelligence introduced decentralized decision-making mechanisms inspired by natural systems. Reinforcement Learning (RL) was presented as a foundational approach, alongside an overview of different RL paradigms and the unique challenges associated with multi-agent reinforcement learning (MARL).

To evaluate RL algorithms for this study, several criteria were considered, including learning efficiency, sample efficiency, exploration strategies, scalability, realism, ease of use, and hyperparameter sensitivity. Given the complexity of multi-agent systems, it is beneficial to control certain variables to better isolate and analyze the impact of specific factors involved in self-organization. Based on these considerations, Proximal Policy Optimization (PPO) and Twin Delayed DDPG (TD3) were selected as the most suitable algorithms for further investigation Based on these factors, Proximal Policy Optimization (PPO) and Twin Delayed DDPG (TD3) were selected as the most suitable algorithms for further investigation.

The next chapter reviews the state of the art in the intersection of stigmergy and MARL, examining how indirect communication through the environment can influence learning and coordination to induce self-organization in multi-agent systems.

3 State of the Art

The pursuit for efficient optimization solutions in complex systems has led to the exploration of various methodologies, including traditional rule-based models [104], game theory approaches, and fuzzy logic systems. However, these methods often struggle to adapt to dynamic changes and fail to fully optimize outcomes in real-time scenarios.

Rule-based systems rely on predetermined sets of rules and heuristics. While effective in certain controlled environments, they struggle to adapt to dynamic changes in environments and fail to optimize in real-time. Game theory based approaches like [105], [106] [107], [108], [109], [110], [111], [112], [113], which model complex systems as strategic interactions among rational agents, are constrained by assumptions of perfect information (full observability), since most real-world problems are partially observable (see section 2.9.2).

Moreover, fuzzy logic systems like [114], [115], [116] which employ linguistic variables and fuzzy rules to model decision-making processes, face challenges in accurately capturing the nuanced interactions and behaviors.

In contrast, Reinforcement Learning (RL) algorithms have emerged as a promising paradigm for optimization [117–119], leveraging principles of learning and adaptation to dynamically adjust strategies based on feedback and environmental cues. By treating optimization problems as sequential decision-making tasks, RL algorithms can continuously refine their strategies to improve outcomes.

RL's adaptability and learning capabilities make it well-suited [120] for addressing the complexities of real-world optimization challenges, where traditional methods often fall short. As optimization challenges evolve, the limitations of conventional approaches become more apparent.

While RL has been primarily applied in contexts such as Traffic Signal Control (TSC) [121], there is growing interest [122, 123] in extending its capabilities to more complex scenarios through Multi-Agent Reinforcement Learning (MARL). MARL extends traditional RL approaches to scenarios where multiple agents interact within a shared environment, offering promising solutions for holistic and adaptive optimization strategies.

One of the key challenges in applying MARL to optimization problems is non-stationarity (section 2.9.3), which can be addressed through mechanisms for coordination and communication among agents. By fostering cooperative behavior and information exchange, MARL systems can adapt to dynamic changes in the environment and optimize outcomes more effectively.

In this review, we explore the state-of-the-art work done in Swarm Intelligence (SI) and MARL, applications of MARL in various areas of traffic management, the mechanisms for coordination and communication, and decentralized control strategies that foster self-organization. Highlights of the related state-of-the-art work is presented in Table 3.1.

3.1 Limitations of Existing Approaches and Motivation for PERL

Many works, such as [124] and [125], successfully demonstrate some form of self-organization through pheromone-inspired mechanisms. However, they are limited to fully cooperative environments, which makes them unsuitable for real-world traffic, where vehicles must balance cooperation (e.g., platooning, merging) and competition (e.g., lane changing, intersection priority).

Other works, such as [127] and [132], explore competitive routing strategies using pheromones. These approaches, inspired by Ant Colony Optimization (ACO), allow vehicle agents to dynamically reroute and decongest traffic. However, they are either rule-based ([132]) or rely on a centralized architecture ([130]), limiting their ability to generalize across varying traffic scenarios.

A fundamental challenge among most existing approaches is that they are not simultaneously decentralized, simple to implement, and interpretable for the chosen problem domain,

Table 3.1: Comparison of Self-Organization and MARL Approaches

Work	Summary	State Space	Action Space
[124]	Self-organization in MARL via stigmergic communication using pheromones in a particle environment.	Discrete	Discrete
[125]	PooL: pheromone-inspired communication using standardized Q-values for MARL in a particle environment.	Discrete	Discrete
[126]	Pheromone-based communication framework using Navier-Stokes equations for robot collision avoidance (Gazebo).	Continuous	Continuous
[127]	Repelling pheromone (inspired by ACO) for vehicle routing, reducing trip duration and fuel consumption.	Discrete	Continuous
[128]	PCDQN: combines DQN with stigmergic communication in a partially observable environment for minefield navigation.	Discrete	Discrete
[129]	ACO-inspired stigmergic rule-based algorithm combined with hierarchical MARL for scalability.	Continuous	Discrete
[130]	MARL vehicle routing for CAVs using ACO-inspired pheromones in SUMO; reduces trip time.	Discrete	Continuous
[131]	MARL with stigmergic communication for emergent platooning in vehicles.	Continuous	Discrete
[132]	CAV routing in SUMO using inverse pheromones inspired by ACO to distribute traffic and reduce congestion; no learning.	Discrete	Continuous
[133]	Independent RL (IRL) in a multi-agent setting using stigmergic communication for collaboration.	Continuous	Discrete
PERL	**Integrates zone-based pheromone in MARL to induce self-organization in environments, using pheromone-dependent reward signals and pheromone-enhanced states.**	**Continuous**	**Discrete**

which involves a mixed environment of cooperation and competition.

- **Decentralization:** Some methods ([128], [130]) incorporate centralized learning, creating bottlenecks that hinder the potential for real-world scalability.
- **Simplicity:** Works such as [126] and [125] introduce complex CNN-based processing pipelines or require advanced fluid dynamic modeling ([126]), making real-world deployment challenging.

- **Interpretability:** Several approaches ([128], [133]) leverage black-box reinforcement learning models, making decision-making opaque and harder to analyze for real-world adaptation.

To illustrate these gaps, Table 3.2 presents a comparative analysis of existing approaches based on key criteria relevant to the problem domain.

Table 3.2: Comparison of Existing Approaches Based on Key Criteria

Work	Type of Environment	Decentralized	Simple	Interpretable	Observability
[124]	Coop	✓	✓	✗	P
[125]	Coop	✓	✗	✓	P
[127]	Comp	✗	✓	✗	F
[132]	Comp	✓	✓	✗	F
[128]	Coop	✗	✗	✗	P
[126]	Coop	✓	✗	✗	P
[129]	Coop	✓	✓	✗	F
[130]	Comp	✗	✓	✗	F
[131]	Coop	✓	✓	✗	P
[133]	Coop	✓	✓	✗	P
PERL	**Mixed**	✓	✓	✓	**P**

Key: Coop = Cooperative, Comp = Competitive, Mixed = Both, F = Full, P = Partial

Given these limitations, to take a step in the direction of MARL in real-world mixed environments, there is a need for an approach that:

- Supports both cooperative and competitive behaviors, reflecting real-world traffic conditions.
- Is fully decentralized, allowing each agent to learn independently without bottlenecks.
- Is simple to implement provided domain knowledge is available to some extent.
- Ensures interpretability, allowing insights into the decision-making process.

This motivates the development of the Pheromone-Enhanced Reinforcement Learning (PERL) algorithm, which integrates pheromone-based stigmergic communication with decentralized MARL in mixed environments.

3.2 Swarm Intelligence and MARL

[124] draws inspiration from the natural phenomenon of Slime mold aggregation, a self-

organizing concept observed in biology. They leverage the local pheromone-based communication used by mold cells to collectively organize into a single organism, particularly in times of food scarcity. This biological concept serves as the foundation for their approach, which integrates it into an Independent Q-Learning algorithm. In their algorithm, the state space is defined as the cell patch exhibiting the highest concentration of pheromone deposited by individual slime agents. This choice of state representation enables the algorithm to capture the collective behavior of the slime mold effectively. To ensure effective learning, they carefully design the reward function, taking care to avoid directly encoding the learning objective of moving the agent towards the highest concentration of pheromone. Their work proves that pheromone-based communication can be learnt. [125] introduces a pheromone-based communication framework called PooL that draws inspiration from pheromone-based systems, incorporating standardized Q-values from the output of the policy network (acting as pheromone) into the state representation. This framework is evaluated across multiple competitive particle environments, including predator-prey scenarios. Agents in this framework perceive a pheromone field, which is characterized by three key domains: the Influence Domain, representing the range within which pheromones released by agents can exert influence; the Observation Domain, indicating the partial observation of the agent within the environment; and the Perception Domain, defining the range of pheromones that agents are capable of perceiving. Information processing involves the use of Convolutional Neural Networks (CNNs) for extracting relevant information, while pheromone reception is facilitated by another CNN dedicated to pheromone perception. Additionally, a virtual medium is employed for the propagation of pheromones.

[126] introduces the PhERS (Pheromone for Every RobotS) framework, which aims to augment the versatility of pheromone-based communication in the field of robotics. The PhERS framework is built upon a sophisticated understanding of pheromone dynamics, leveraging principles from fluid dynamics, particularly the Navier-Stokes equation. By incorporating factors such as wind effects, evaporation, diffusion, and injection impact size into their model, the authors provide a comprehensive framework for simulating and analyzing the behavior of pheromones in robotic environments. To evaluate the performance of their framework, they conducted experiments in a Gazebo robot collision avoidance environment,

featuring both static and dynamic obstacles. The results of these experiments highlight the superior performance of their Proximal Policy Optimization (PPO) controller compared to traditional hand-tuned controllers. [127] investigated a decentralized green traffic optimization framework, integrating swarm intelligence principles into Connected and Autonomous Vehicles (CAVs) to mitigate congestion. This framework introduced dynamic traffic routing methods based on Ant Colony Optimization (ACO) as a technique for traffic routing optimization. Modeled after real ants' behavior, ACO leverages digital pheromones to guide vehicles dynamically. Notably, the study proposed the concept of repelling pheromones that push vehicle agents away from each other for self-organization, thereby reducing congestion and improving overall traffic flow. Their evaluations suggest significant improvements in fuel consumption, emissions, and trip duration.

[128] introduces a novel communication framework termed Pheromone Collaborative Deep Q-Network (PCDQN), which integrates DQN with a stigmergy-based approach driven by pheromones. Unlike conventional methods that rely on predefined communication protocols or additional decision modules, PCDQN harnesses stigmergy as a dynamic communication mechanism among independent reinforcement learning agents. Through experimentation in partially observable environments, particularly in minefield navigation tasks, PCDQN demonstrates superior learning productivity for multiple agents compared to standard Deep Q-network (DQN) approaches. [129] combined ACO-inspired stigmergic rule-based algorithm with hierarchical MARL for scalability in box environment, where agents must collaborate to achieve common objectives, such as object transportation and obstacle navigation. Hierarchical MARL added another layer of organization to the system by structuring agents into hierarchies or layers. Each layer had its own set of objectives, with higher layers overseeing the actions of lower layers. The work employed a diverse array of pheromones to facilitate communication and collaboration among agents, including a distance pheromone for information regarding the distances between different nodes and goal nodes within the environment, a box pheromone that provided insight into the presence and state of boxes within the environment, a hole pheromone that conveyed information about the presence of openings or holes to avoid within the environment and an exploration pheromone that incentivized agents to explore new areas within the environment.

Another ACO-based vehicle routing work [130] employs a variety of pheromones including traffic density pheromone, road length pheromone, travel time pheromone and colored pheromones. Each Connected Vehicle (CV) is treated as an ant that leaves pheromone traces along its path. To differentiate between different traffic flows, a concept called colored CVs is introduced. CVs with the same destination leave a specific type of pheromone, known as colored pheromone. Only CVs heading towards the same destination can interact with this colored pheromone, allowing for the distinction of multi-source multi-destination traffic flows during the routing process. Their evaluations in SUMO show reduced trip time when compared with shortest path routing. [131] is an interesting work that introduces stigmergic communication in MARL for emergent platooning in vehicles. They adopt the conventional messages used in platooning like Gap and Maneuver (Join and Exit) messages for pheromones. Empirical evaluation shown indicates that the emergent Join and Exit maneuvers using pheromones has better recovery from disturbances in the string stability of the platoon.

Similar to the work seen in [127], [132] shows a distinction between traditional ACO-based routing methods and Inverse Pheromone-based routing (IPR). The conventional method guides the vehicles to move to the road segments with perceived lower travel time. The problem arises when many vehicles move to the same road segment at the same time, because now that new road segment is observably congested. Instead, in [132], the IPR method detects congested segments by using occupancy rate, then reroutes vehicles to alternative path with a lower inverse pheromone value. There is no learning involved in IPR, and the evaluation results were slightly mixed, but overall the IPR method appeared to have better average trip duration times. In the work of [133], Stigmergic independent reinforcement learning (SIRL) is introduced as an algorithm for IRL with stigmergy in a pixel particle environment. They divide the process of stigmergy into 4 components, namely, medium, trace, action and condition. It is implied that the medium is the one holding this pheromone-based information. They employ a federal training method, which means that each agent adjusts the weight of its neural network from both its own experience and the policy weight update gradients from other agents. They evaluate the results on a particularly interesting application of particles; to form desirable shapes like numbers or letters. The results indicate comparable level of performance to another similar method.

There are also some highly notable older approaches. [134] introduces Phe-Q, a novel adaptation of Q-learning that incorporates pheromone-based communication to enhance learning in multi-agent environments. This approach draws inspiration from stigmergic systems in nature, particularly ant foraging behavior, where pheromones serve as a decentralized means of communication. In traditional Q-learning, agents update their value functions based on direct environmental feedback, but Phe-Q extends this by allowing agents to deposit and perceive artificial pheromones that encode useful information about past experiences. These pheromone traces provide an additional layer of guidance, influencing exploration and exploitation decisions dynamically. Unlike conventional approaches that rely solely on immediate rewards, Phe-Q leverages these environmental markers to shape long-term strategy formation, promoting more effective coordination among agents. The authors evaluate their approach in a multi-agent grid world environment, demonstrating that Phe-Q accelerates learning convergence by reinforcing successful trajectories while reducing inefficient exploration. Their results highlight that pheromone-based communication enables agents to collectively adapt to dynamic environments more efficiently than standard Q-learning.

Expanding upon their initial work, [135] presents an improved version of Phe-Q, refining the role of synthetic pheromones in reinforcement learning. In this enhanced approach, pheromone signals not only guide agent movement but also undergo controlled decay, ensuring that older, potentially obsolete information does not mislead decision-making. The introduction of pheromone reinforcement mechanisms enables agents to amplify useful signals while diminishing the influence of less relevant ones, thereby optimizing the balance between exploration and exploitation. The updated framework also integrates a more structured approach to pheromone diffusion, preventing excessive environmental saturation that could lead to suboptimal decision patterns. Through a series of experimental evaluations, the authors demonstrate that the improved Phe-Q achieves superior performance in multi-agent tasks, particularly in scenarios requiring adaptive coordination and decentralized problem-solving. Their findings suggest that synthetic pheromone-based reinforcement learning provides a viable alternative to traditional methods, particularly in environments where direct agent-to-agent communication is limited or impractical. The implications of

this work extend beyond simple multi-agent coordination, offering insights into how bio-inspired principles can enhance autonomous decision-making in complex, dynamic systems.

3.3 Cruise control

Cruise control systems, particularly adaptive cruise control (ACC) is about the control over acceleration or deceleration of a vehicle. [136] conducted experiments to assess the string stability of seven 2018 model year ACC-equipped vehicles, providing insights into the behavior of these systems in car-following scenarios. Building on this work, [137] developed a data-driven approach to calculate string stability using field data from a 2015 luxury electric vehicle, further enhancing understanding of ACC systems. Towards safety-critical control, [138] revisited the classic Adaptive Control Lyapunov Functions (aCLFs) formulation in the context of ACC, demonstrating the framework's ability to achieve stability and safety in simulations. [139] proposed a personalized ACC system based on driving style recognition and model predictive control (MPC), highlighting the importance of catering to individual driving preferences while optimizing car-following performance. [140] focused on measuring the response time of ACC-enabled vehicles in car-following conditions, shedding light on the practical implications of these systems on road capacity.

[141] addressed the challenge of synthesizing string-stable controllers for cooperative adaptive cruise control, highlighting the importance of choosing appropriate control parameters for ensuring system stability. [142] introduced a multi-agent reinforcement learning approach for networked system control using a novel reward function and parameter sharing, showcasing the potential of decentralized control policies in scenarios such as adaptive traffic signal control and cooperative ACC. Finally, [143] compared Deep Reinforcement Learning (DRL) and Model Predictive Control (MPC) for ACC design, offering insights into the effectiveness of different control methodologies in car-following scenarios.

3.4 Collision Avoidance

We can think of collision avoidance as another function of cruise control. Apart from ensuring the most efficient possible acceleration in vehicles, it must enforce a strict no-

collision policy. They go hand-in-hand. In the context of aerial traffic control, [144] proposed a collision avoidance method based on DDPG to ensure conflict-free conditions for aircraft. Similarly, [145] presented a decentralized on-demand airborne collision avoidance framework for urban air mobility, allowing unmanned aircraft systems (UAS) to independently plan and execute missions.

Back on the ground,[146] developed a new safe lane change trajectory model and collision avoidance control method for automatic driving vehicles, while [147] demonstrated the effectiveness of trained controllers for collision avoidance in highway traffic merging scenarios using deep multi-agent reinforcement learning. In the domain of autonomous vehicles, [147] proposed a risk assessment-based decision-making algorithm for collision avoidance in multi-scenarios, and [148] presented an inter-vehicular communication strategy for avoiding critical collisions on Indian roads. [149] proposed a fuzzy-logic based intelligent conflict detection and resolution algorithm for collision avoidance in multiple ship encounters in marine traffic.

Reference	**Summary**
[150]	Reinforced Cooperative Autonomous Vehicle Collision AvoidancE (RACE); novel Co-DDPG algorithm; relative distances instead of locations; 65% less collisions in varying traffic densities
[151]	Multi-agent centralized control of CAVs for travel efficient and safe car-following
[152]	Multi-agent A3C for cruise control to simulate varied aggression human driving

Table 3.3: Highlights of learning collision avoidance

There is also some work on bio-inspired collision avoidance. Some of these strategies using virtual pheromones have been explored in the context of swarm systems, with [153] proposing a bio-inspired collision avoidance strategy using virtual pheromones for autonomous vehicles operating together on roads. Finally, [154] presented an environment-centric navigation policy that learns a set of traffic rules to coordinate a large group of unintelligent agents with basic collision-avoidance capabilities. Overall, we note the diverse applications of RL in collision avoidance across air traffic control, autonomous driving vehicles, marine traffic, and swarm systems.

3.5 Lane-changing

[155] applied a DQN-based method combined with rule-based constraints for autonomous driving lane change decision-making tasks. [156] introduces the 'CM3' architecture for multi-goal cooperative lane changes in SUMO by combining function approximation and cooperative actions. A harmonious lane changing approach [157] was presented in MARL, where agents exchange strategies to reach a zero-sum game state.

[158] proposed a hierarchical reinforcement learning approach for multi-agent cooperation, focusing on high-level decision-making and low-level individual control by decomposing an overall cooperation task into a hierarchy of discrete sub-tasks. For example, a high-level lane-change operation consisted of a slow-down, an accelerate and a lane-change low-level actions. [159] revisited the lane-change problem and introduced a bi-level lane-change behavior planning strategy using MARL with right-of-way collaboration awareness.

Reference	**Summary**
[160]	MARL with vehicles as agents; information from a vehicle acting as Road Side Unit (RSU) for local and global performance
[161]	Multi-agent advantage actor-critic (MA2C) method; multi-objective reward design and parameter sharing for safe and comfortable lane-changing
[162]	Lane change process subdivided into lane-change process in original lane and subject lane; evaluated in MATLAB
[163]	High-level decision making using Dynamic Coordination Graphs for multi-lane lane-changing and over-taking in highway scenario
[164]	Centralized multi-agent PPO control for lane-changing maneuvers in highway scenario

Table 3.4: Highlights of learning vehicle lane-changing

Among the recent research in lane-changing vehicles using MARL, [160] claims to be the state-of-the-art in all aspects of traffic flow and efficiency. They use information from the Road Side Unit (RSU) to improve traffic flow in a highway scenario locally and globally. [165] formulated lane-changing decision-making for AVs in mixed-traffic environments as a MARL problem, considering neighboring AVs and Human-Driven Vehicles (HDVs). [166] addressed high-conflict driving scenarios requiring negotiations between agents with equal rights and priorities.

3.6 Platooning

From [13], in platooning, a group of vehicles follow a certain platoon leader and attempt to maintain similar vehicle state properties. For example, they maintain the same velocity and unnecessary lane-changing is minimized unless required. A platoon of vehicles could potentially rearrange itself like a flock of birds. So we can think of platoons as smaller local self-organizing groups.

In [167]'s highway setting, by constraining platoons to a special lane, they reduce inter-platoon interactions with other vehicles so that platoon level control changes can be made without disturbing the rest of the flow. The first decision that is taken is which agent from the system should merge into a platoon, and the second control decision for each agent is the selection of the platoon that it wants to join and from which side. Again, this shows decent results but does not scale as well as a completely decentralized approach would with the availability of CAV technology now. In [168], a real-world scenario is simulated to reduce fuel consumption and travel time in different levels of congestion. It uses a Predictive cruise control (PCC) algorithm based on information collected from the vehicle in front of it (position, velocity, acceleration) and the Signal Phase and Timing message (SPaT) from the infrastructure through V2V and V2I technology respectively to enable reference velocity of the platoon to be set in such a way that the platoon does not have to stop at the intersection. The vehicle platoon leader is controlled at the intersection to reduce travel time and fuel consumption. Communication is unidirectional. Though there is cooperation by learnt communication among the vehicles within the platoon in [169], there is no coordination between the different platoons that use a MARL-ACC controller based on REINFORCE to maintain the velocity of vehicles in a platoon. Only a platoon of 3 vehicles and 5 vehicles was used for evaluation, which was also not realistic due to discrete state and action spaces.

Platooning approaches, as outlined in Table 3.5, encompass a spectrum ranging from decentralized to centralized methodologies. At the heart of platooning behavior lie the pivotal maneuvers of joining, exiting, and maintaining safe gaps within the convoy. These maneuvers serve as fundamental building blocks.

To sum up, the missing links in all the approaches discussed above come from continued

Reference	Summary
[167]	Multi-agent centralized coordination of vehicles desiring merging with a platoon and decentralized vehicle lane-changing optimization
[168]	Coordination between platoon and intersections based on V2I and V2V communication to enable non-stop movement of platoon through intersection
[169]	Learning communication between platoon vehicles in a mixed-highway scenario for cooperative cruise control

Table 3.5: Highlights of learning platooning in vehicles

use of centralized controllers, unrealistic evaluation scenarios or simulation software, lack of effective communication explicitly for coordination, unsuitable state representations or rewards and even unrealistic discrete actions instead of continuous actions.

3.7 Intersection management

When we talk about traffic, one of the first visuals that pop into mind is that of traffic at intersections or junctions. The management of traffic at intersections, characterized as UTC is a significant challenge in traffic management. [170] proposed a Q-Learning approach for TSC at urban intersections. Similarly, [171] proposed a centralized coordination scheme of CAVs at an intersection without traffic signals using PPO to improve computation efficiency when compared to model predictive control (MPC). [172] focused on controlling traffic flow within a network of multiple signalized intersections with traffic ramps using Q-learning, which provides flexibility to change traffic signals according to traffic conditions. [173] also proposed a traffic signal control system using Q-learning to maximize the number of vehicles crossing an intersection and balance signals between roads.

In addition to TSC, forecasting traffic flow at intersections is crucial for advanced trip planning and traffic management. [179] developed a two-layer stacking model for intersection short-term traffic flow forecasting by integrating different modeling methods from Machine Learning (ML) along with RL. The integration of RL and ML techniques has shown potential to further improve the performance of traffic signal controllers. [180] focused on real-time intelligent autonomous intersection management using RL to adjust time and speed for collision-free passage through intersections.

Reference	Summary
[174]	Attention mechanism in Multi-agent Actor-Critic networks to enhance arterial traffic information extraction
[175]	Multi-Objective Multi-Agent DDPG with locally rewarded and globally rewarded agents using age-decaying weights
[176]	SocialLight learns cooperative policies by assigning credit to agents locally
[177]	Uses Max-Pressure concepts to enhance state for TSC, combined with graph attention mechanism (GAT) to learn state representation
[178]	MARL with spatio-temporal feature fusion; correlations between timesteps and between subnetworks are taken into account in the state

Table 3.6: Highlights of learning intersection management or junction control

3.8 Motivation

Existing studies demonstrated previously, the potential of RL in traffic management, but they often overlook the complex interactions and emergent behaviors that arise from the coordination of decentralized agents. Traditional optimization methods and RL techniques have shown limitations in optimizing performance in partially observable and non-stationary environments.

To address these limitations, there is a need to explore innovative approaches inspired by swarm intelligence concepts like stigmergy, which induce emergent behavior through self-organization. By leveraging principles of self-organization and collective intelligence, MARL algorithms can facilitate decentralized coordination among agents. In the context of traffic, integrating MARL techniques with advanced sensing and communication technologies from Connected and Autonomous Vehicles (CAVs) can enable real-time information exchange and decision-making among decentralized CAV agents.

Building upon existing research, our work is motivated by the use of digital pheromones that leverage stigmergic mechanisms from Ant Colony Optimization (ACO). We aim to induce global self-organizing behavior from the local interactions of Multi-Agent Reinforcement Learning (MARL) agents. We will evaluate this approach in multiple traffic scenarios that exhibit both partial observability and non-stationarity.

4 Methodology

In this chapter, we delve into the proposed methods for integrating MARL with stigmergy mechanisms from nature. We choose traffic as our problem domain for evaluation, because traffic environments are complex systems characterized by partial observability and non-stationarity. Throughout this chapter, we aim to elucidate the rationale behind our design choices, introduce the key components or concepts underpinning the design, and elucidate its functionality alongside any references to substantiate its validity.

4.1 Requirements for the self-organizing system

To reiterate, the essentials required for a system to be self-organizing [9] as mentioned in chapter 2 are as follows:

1. **Positive Feedback**: Positive signals reinforces actions or states perceived as favorable or beneficial. This feedback signal communicates to entities within the system that certain behaviors or conditions are conducive to system stability or performance enhancement.
2. **Negative Feedback**: Discourages undesirable actions or states. By communicating the undesirability of certain behaviors or conditions, negative feedback mechanisms act as a regulatory force. Negative feedback discourages deviations from desired equilibria and encourages the exploration of alternative states or arrangements. For example, if multiple vehicles decide to move to a lane with high pheromone concentration, that lane is now congested and negative feedback will help repel some of these vehicles to other lanes.
3. **Amplification of Fluctuations**: Fluctuations refer to random perturbations caused

by the system's entities or interactions. These create a cascading effect of one entity's behaviour over another. These fluctuations can compound over time, creating amplified feedback signals that push emergence of complex patterns in self-organizing systems. An example is the cycles of population growth and decline in rabbit and fox populations, driven by fluctuations in predator-prey interactions.

4. **Multiple Interactions**: The presence of multiple interactions among entities is a critical component. This entails a minimal density of entities, allowing them to leverage the experiences and behaviors of others. A simple example are study groups where individuals learn from each other's experiences and multiple interactions facilitates collective learning within the system. This process enables entities to gain insights, strategies, and feedback from their peers, contributing to the system's overall adaptability and resilience.

These components together foster the proliferation of beneficial behaviors or arrangements, driving the emergence of stable equilibria in a system. These emergent behaviours do not have to be cooperative, but they are usually the ones that stand the test of natural selection [9].

4.2 Self-organizing system design

A characteristic feature of self-organizing systems is the emergence of complex global patterns from multiple local interactions in the system [74]. This theory was further supported by [133], stating that stigmergy [181] can serve as a potential solution to the decomposition of a global objective. We take inspiration from Dorigo's ACO [11] that demonstrates the potential for bio-inspired algorithms to solve challenging optimization problems.

We propose to leverage stigmergic communication via digital pheromones to induce self-organization in our MARL environment, guiding the emergent behavior to a robust global attractor state (through evaporation and diffusion mechanisms) that aligns with agent objectives through appropriate feedback signals. The various components of our system and the underlying processes involved are:

- Pheromone Field

- Zones
- Pheromone perception range
- Pheromone design and computation
- Stigmergic processes
  - Evaporation
  - Diffusion

4.2.1 Pheromone Field

In nature (as explained previously in Chapter 2), ants deposit chemical pheromone [182] trails along their paths when searching for food. These pheromone trails serve as a form of communication, allowing ants to exhibit a phenomenon known as stigmergy [181], wherein their actions modify the environment, influencing the behavior of other ants. As ants find food and return to the colony, they reinforce the pheromone trails, making them more attractive to other ants. The shorter trail, being shorter, is more likely to accumulate a higher concentration of pheromone since ants can traverse it faster, resulting in a higher rate of pheromone deposition compared to evaporation (see Figure 2.3 in section 2.5.1). This positive feedback loop causes the pheromone trails to intensify over time, guiding more ants towards the shortest trail to the food source. We will emulate this pheromone behavior for our CAV agents, encoding useful information as detailed later in Section 4.2.5.

The pheromone field provides a comprehensive view of all pheromone concentrations throughout the system, with agents accessing only the portions relevant to their local surrounding. It operates as a virtual representation of pheromone concentrations across designated spatial regions called zones (see section 4.2.2) within the environment. The primary role of the pheromone field is to manage parallel computational updates, such as pheromone deposition, evaporation and diffusion, to prevent race conditions between zones.

The initialization process involves generating the pheromone field from a random uniform distribution. With a mean of 0 and a standard deviation of 1, each zone in the field is initialized with a random number from this distribution. The selection of a uniform distribution ensures that each value within the distribution has an equal probability of being chosen, for different random seeds and experimental replicability.

4.2.2 Zones

A zone is a predefined local spatial region on the environment which maps onto some value of pheromone concentration in the pheromone field. In our implementation, we discretize the pheromone field into zones rather than using a function approximator that deals with continuous values for the sake of:

- **Simplicity and Efficiency:** Discretizing the environment into zones simplifies the pheromone update mechanism and reduces computational overhead [11, 12]
- **Interpretability:** Discretization often enhances the interpretability of algorithms by providing a more structured and understandable representation of the environment. In RL, discrete representations facilitate easier interpretation of learned policies and behaviors [5].
- **Robustness and Generalization:** When data is discretized, it tends to smooth out variations and fluctuations in the original continuous data. By aggregating information within discrete zones, algorithms can better generalize across similar states [183].

Each zone maintains its unique pheromone levels, updated by the actions of agents local to that zone. It is through these zones in the pheromone field that we realize the process of pheromone deposition by local CAV agents, evaporation and diffusion into surroundings. The size of these zones are predefined, often selected for interpretability of agent behaviour. To optimize network connectivity and minimize potential inconsistencies in pheromone representations, we define smaller zones (ranging from 200 to 300 meters) within the broader 5G-NR [184] framework, despite its capability to span up to 5000 meters. This approach reduces the likelihood of vehicles being out of range from each other, ensuring more consistent pheromone data until they re-enter each other's range.

In our highway scenarios, these zones are discretized spaces along the length of the lanes. For example, if the number of zones on a 3-lane 1000m road is 4, then each lane on the road is divided into 4 parts, of length 250m each. In intersection scenarios, each leg of an intersection is considered a zone.

4.2.3 Pheromone perception range

The environment being partially observable means that agents would not have full visibility of the state, or the pheromone field. The pheromone perception range describes the number and location (or direction) of zones, for which the agent is able to perceive pheromone concentrations in at any given time. Limiting the perception of pheromones (partial observability) enables the investigation of local interactions of agents with the pheromone field in the environment leading to an emergence of system-wide behaviour.

In lane-changing scenarios, the perception range of pheromones is set to the current zone, and immediate adjacent zones on the left and right of the CAV agent. In intersection scenarios, for an agent at the incoming leg of any intersection, the perceived pheromones include the pheromone concentration on its own leg, those in the adjacent incoming leg, and the 2 other outgoing legs. This means that an agent on the outgoing leg does not perceive any pheromones other than its own. However, in an interconnected intersections scenario with multiple intersections, we can see that the agent on the outgoing leg of an intersection can also be considered on the incoming leg of an adjoining intersection.

4.2.4 Pheromone design

Ants use stigmergy (section 2.5.1) to find the shortest path to food. They do this by following pheromone trails that have higher concentration. Since we want to integrate MARL with digital pheromones, we can use it as an additional learning signal to guide the policy of the agent to self-organize.

The mean absolute deviation (MAD) [185] is a measure of the average absolute deviation of a set of values from their mean. It quantifies the average distance between each data point and the mean of the data set, regardless of the direction of the deviation.

We use a novel feedback signal that scales inversely with MAD as $e^{-}\beta\text{MAD}$. β is the MAD multiplier, set mostly to 1 arbitrarily for all experiments. This ensures that the reward is highest ($+1$) when the local pheromone levels are perfectly equalized, i.e., when $\text{MAD} = 0$. In the example (Figure 4.1), we see a lane changing scenario where CAV agents are moving from left to right on a 3-lane highway. The perceived local pheromones of the ego agent

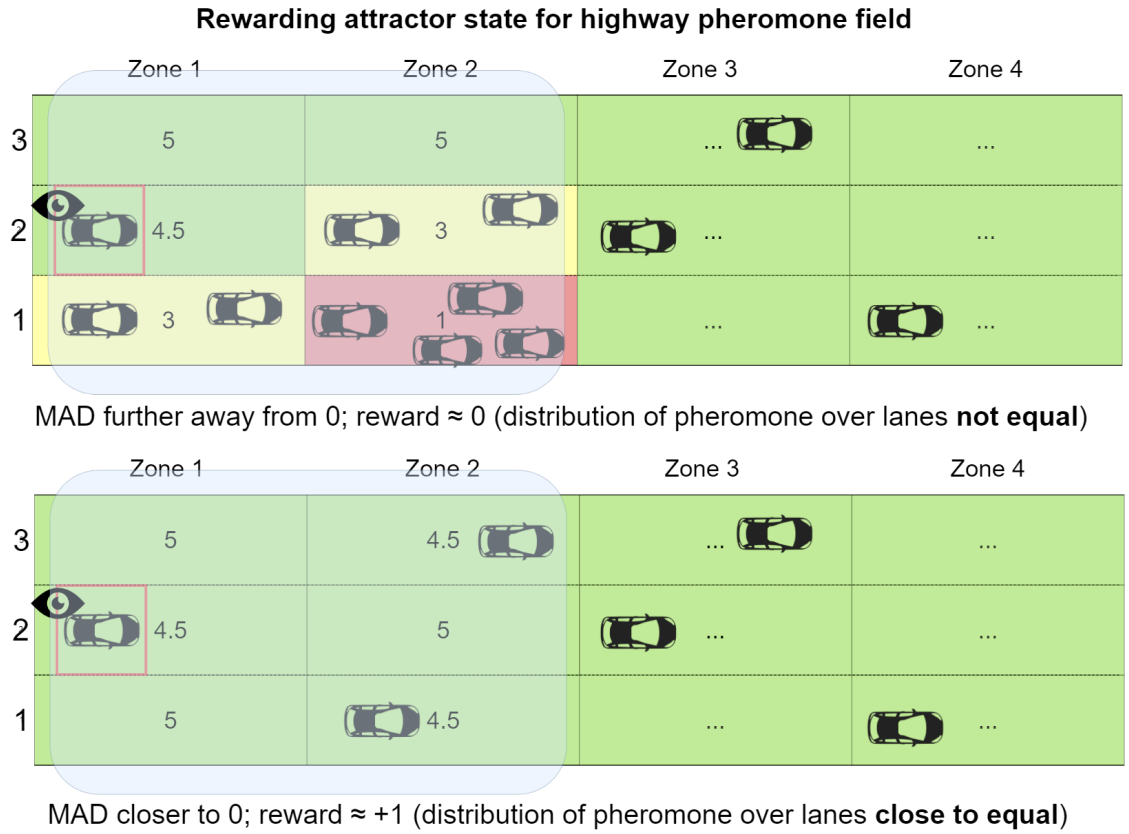

Figure 4.1: Mean absolute difference (MAD) of pheromone levels in the pheromone field for feedback signal

captures pheromone concentrations in its current zone, the immediately adjacent zones, and all the immediate upcoming zones in each lane. The closer the pheromone concentrations are to each other, or the more equalized they are, they higher the MAD reward.

In our other example (Figure 4.2), each leg of the intersection acts as a zone. In this case, the rewarding of distribution of pheromone propagates to other connected intersections due to the process of diffusion. These ACO-inspired digital pheromones act as a conduit for indirect communication among agents operating within the decentralized system. Hence, the feedback received by agents for equalizing pheromones in their local vicinity translates to a global behaviour that strives to equalize pheromone levels in all zones. This leads to a emergent behaviour that coordinates equal redistribution of traffic.

Nash equilibrium

In multi-agent systems, a Nash equilibrium represents a stable state in which no agent can unilaterally improve its outcome by changing its strategy [53]. In this context, Nash

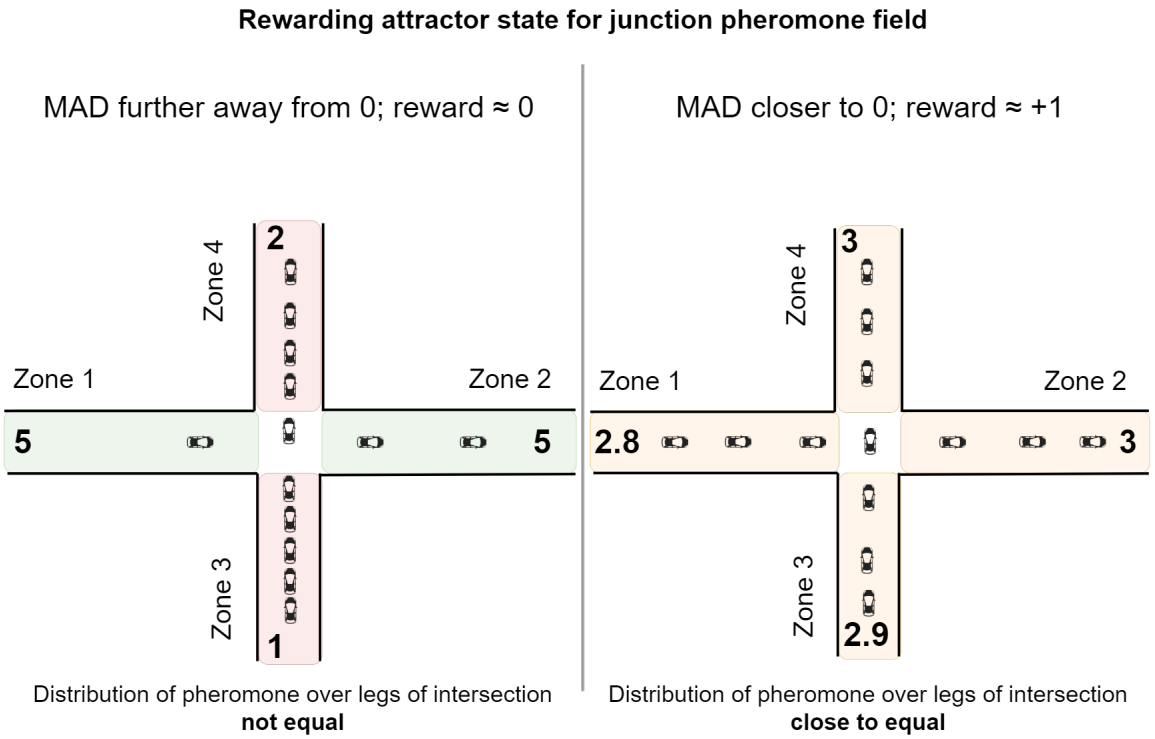

Figure 4.2: Mean absolute difference (MAD) of pheromone levels in the pheromone field for feedback signal

equilibrium would correspond to a scenario where each vehicleâs decision-making process results in an equilibrium state where no single vehicle benefits from deviating from its learned policy, assuming all others continue following theirs.

However, computing Nash equilibria explicitly in this system presents several challenges. The continuous state spaces, the stochastic nature of vehicle arrivals, and the decentralized decision-making structure make analytical derivation of Nash equilibria intractable. Traditional game-theoretic approaches often require simplifying assumptions, such as discrete action spaces or perfectly rational agents, which do not align with the reinforcement learning-based approach used in this study. As a result, no closed-form Nash equilibrium solutions are available.

Despite the inability to explicitly compute Nash equilibria, the learned policies can still be shown to align with the general intuition behind Nash equilibria, where agents reach a steady-state interaction pattern that does not incentivize unilateral deviations, for example, through equalization of pheromone levels on each zone/leg of an intersection.

4.2.5 Calculation of pheromone

To describe digital pheromones, we have described its functionality or any related concepts like the pheromone field and the zones within that field. Now we describe the content of the pheromone in detail. The information that is propagated by the pheromone signal is intended to guide the appropriate actions of an agent as a signal. Our aim is to encapsulate the degree of congestion through these pheromone signals, telling us how good or bad it is to be in a certain zone in the environment (in terms of congestion level).

To quantify, we explored different combinations of features of vehicles that characterize congestion, including speed, maximum speed, acceleration, waiting time, and others [25]. The pheromone also depends on the type of scenario we are applying the digital pheromones for. In lane-changing scenarios, it's essential to accurately represent dynamic blockages through pheromone signals. Two key indicators of reduced road capacity on a lane are the speed and acceleration of vehicles.

Speed serves as an indicator of traffic fluidity [186], while acceleration or deceleration provides insights into vehicles' instantaneous behavior. For instance, deceleration signals the number of vehicles slowing down and the extent of their deceleration, representing the flow of traffic opposing the desired direction. Both in lane-changing and intersection scenarios, we leverage vehicle speed and acceleration to determine pheromone concentration within a zone. However, in intersection scenarios, we also incorporate the waiting time at the intersection as an additional measure of congestion level within the zone.

Features such as agent speed and acceleration contribute positively, while waiting time inversely correlates with heightened pheromone concentration in the surrounding environment.

The combining of features from agents is done using a standardization function (4.1) [187] and min-max normalization (4.2) [188] to bring them to a comparable scale for aggregation or summation.

To elucidate, the purpose of standardization is to rescale the features such that they have the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. Generally, to standardize a feature, you subtract the mean of the feature from each

value and then divide by the standard deviation of the feature. But since we only know the range of values the features can take, these are done using the minimum and maximum.

The standardization formula is given by:

$$z = \begin{cases} \frac{X - \frac{X_{\max} + X_{\min}}{2}}{X_{\max} - X_{\min}} & \text{if } X_{\max} \neq X_{\min} \\ X - \frac{X_{\max} + X_{\min}}{2} & \text{otherwise} \end{cases} \tag{4.1}$$

where:

z is the standardized value of X,

X is the original value,

$X_{\min}$ is the minimum value of the data set, and

$X_{\max}$ is the maximum value of the data set.

The standardized values now have mean 0 and standard deviation 1. To ensure that these standardized values are within a fixed range, we use normalization.

To expand, the purpose of normalization is to rescale the features into a fixed range, usually between 0 and 1. This is useful since the selected features have different units or scales, and we want to bring them to a similar scale. We have opted to choose min-max normalization since we have minimum and maximum available for our data.

The min-max normalization formula is given by:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{4.2}$$

where:

X_{norm} is the normalized value of X,

X is the original value,

$X_{\min}$ is the minimum value of the data set, and

$X_{\max}$ is the maximum value of the data set.

Once we possess a set of standardized and normalized features from each agent, the contribution of each agent is determined by aggregating (summing) these values.

$$C = \sum_{i=1}^{n} \text{Feature}_i \tag{4.3}$$

where:

C is the agent's contribution ,

$Feature_i$ is the standardized and normalized feature from the agent (e.g., speed, acceleration) ,

Considering that multiple agents can contribute to a single zone, we take the average over these individual contributions (4.3) from all agents in a zone. C_{mean} represents the overall pheromone concentration merged into the pheromone field representing the existing digital pheromone in the environment. This incorporation of new signals into the pheromone field is done along with the evaporation and diffusion mechanisms for efficiency.

Moreover, the calculation of pheromone concentration is not a static process; rather, it evolves dynamically over time as agents move and interact within the environment. As agents deposit and sense pheromones, the distribution and concentration of pheromones within the environment change, leading to shifts in agent behavior responsive to changing environments.

4.2.6 Stigmergic Processes

This section elucidates the stigmergic processes entailed in the merging of pheromone information into the relevant part of the pheromone field as perceived by agents. The key processes discussed are evaporation and diffusion. Evaporation aids in the elimination of outdated data, while diffusion facilitates the dissemination of information to the surroundings, fostering global coordination through the phenomena of cascading (discussed in [189]).

Evaporation

In nature, ants use pheromones to communicate and navigate. When an ant finds food, it leaves a trail of pheromones for others to follow. Ants sense these pheromones using chemosensory receptors, similar to smelling in humans. As ants move, they detect pheromone concentrations in the air and follow trails with higher concentrations. However, pheromones naturally evaporate over time, dispersing into the environment. This evaporation process ensures that old trails fade away, encouraging ants to explore new paths and adapt to changing conditions.

Similarly, we adopt this evaporation process (Figure 4.3) as a mechanism that causes the gradual reduction or decay of pheromone trails over time. After being deposited by agents as they traverse the environment, pheromones continue to dissipate with the passage of time. Evaporation is a crucial aspect of our approach, as it helps prevent the accumulation of outdated or misleading information and encourages exploration of new states. Adjusting the rate of evaporation allows for fine-tuning the balance between exploiting well-established paths and exploring new possibilities in the search space. We control this rate of evaporation using the evaporation factor (j).

During the learning process, after we calculate the mean contribution from agents in a zone, a part of the existing pheromone is evaporated before being integrated with the new computed pheromone signal as follows:

$$P' = (1 - j) \times P + j \times C_{\text{mean}} \tag{4.4}$$

where:

P' is the new value of the pheromone,

P is the existing value of the pheromone,

j is the Evaporation factor/coefficient, and

C_{mean} is the mean of contributions of all agents in a zone.

The evaporation rate significantly influences the persistence of pheromone signals, which in turn regulates agent behavior by affecting the durability of pheromone trails. By adjusting the rate at which pheromone signals diminish, we strike a balance between locally preserving critical information and encouraging exploration of new paths. This control is governed by the evaporation factor (j), which ranges inclusively from 0 to 1, covering both ends of the spectrum. Figure 4.3 illustrates the pheromone evaporation mechanism, where, for example, $j = 0.1$ or 10

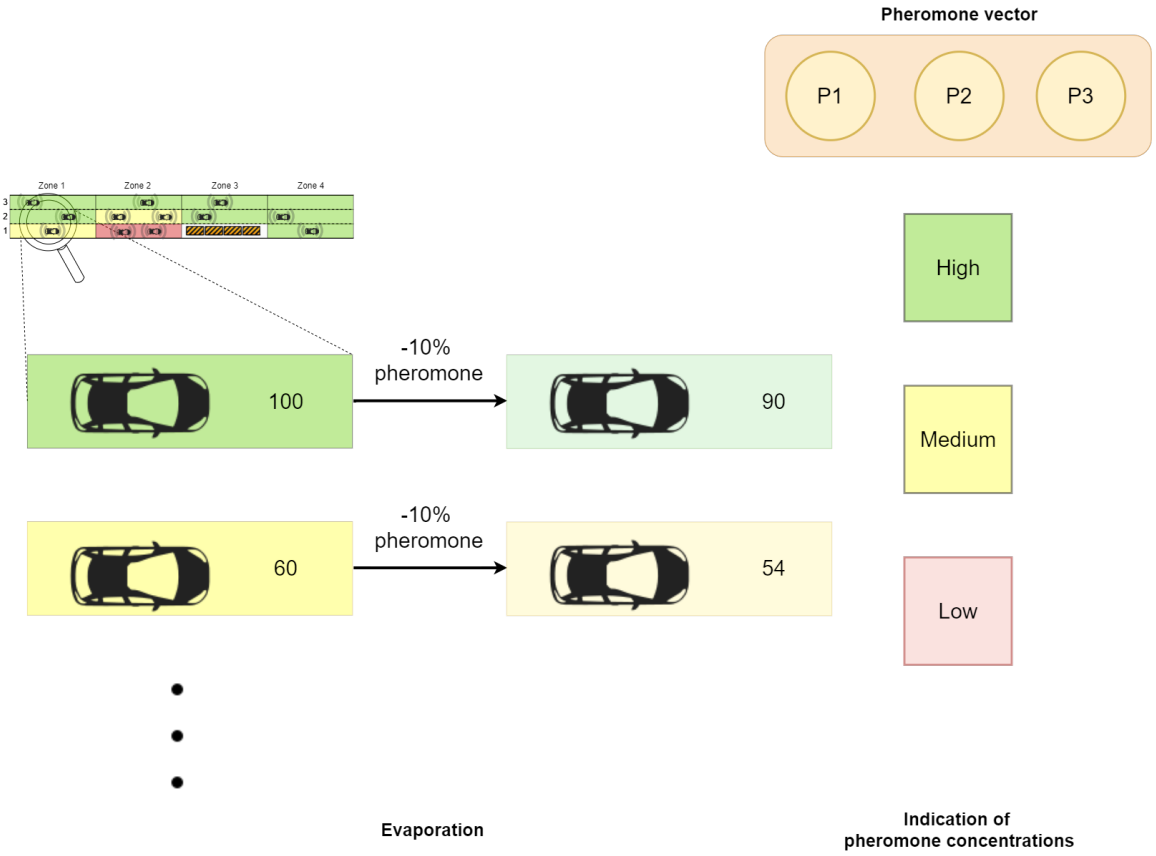

Figure 4.3: Pheromone evaporation mechanism (eg. j = 0.1 or 10%)

An intriguing aspect of evaporation (Equation 4.4) in our approach lies in its nature. As the concentration of pheromone increases, the evaporation process now dissipates a larger portion of the concentration compared to before. This is due to the percentage-based nature of evaporation. Therefore, for pheromone signals associated with favorable actions in specific states to persist, local agents must continually interact with the environment and deposit pheromone in that zone.

Diffusion

In natural systems, diffusion also plays a crucial role in facilitating coordination from local to global scales, similar to how ants utilize pheromones for communication and navigation. Just as ants deposit pheromones to guide their colony members, molecules in a diffusive system spread from regions of high concentration to regions of lower concentration, allowing information to propagate across the system. This spreading of molecules enables the synchronization of behaviors and the emergence of collective actions, promoting coordination among agents at both local and global levels. As molecules diffuse, they contribute to the exchange of information and the alignment of behaviors, ultimately facilitating harmonized responses and adaptive behaviors across the entire system.

A similar diffusion (4.5) process occurs between zones for the purpose of *spreading* the pheromone signal to surroundings zones:

$$P_{\text{affected}} = (1 - k) \times P_{\text{affected}} + k \times P_{\text{influencer}} \tag{4.5}$$

where:

$P_{\text{influencer}}$ is the value of the pheromone in the influencing zone,

P_{affected} is the value of the pheromone in the affected zone,

k is the Diffusion factor/coefficient, and

This diffusion happens between the zones of the pheromone field, computed in parallel to avoid race conditions. In a highway setting, anterior zones are regarded as influencing zones for posterior zones, which are impacted by alterations occurring ahead of them. For example, a vehicle on a the freeway will be affected by any disturbances on the road in front of him, not behind him. So the zones in front will affect the zones in the back.

Conversely, within intersection scenarios, incoming zones influence outgoing zones, and any changes in the conditions of incoming zones impact the dynamics of outgoing zones.

Our goal is to accurately capture fluctuations in congestion levels through our pheromone model.

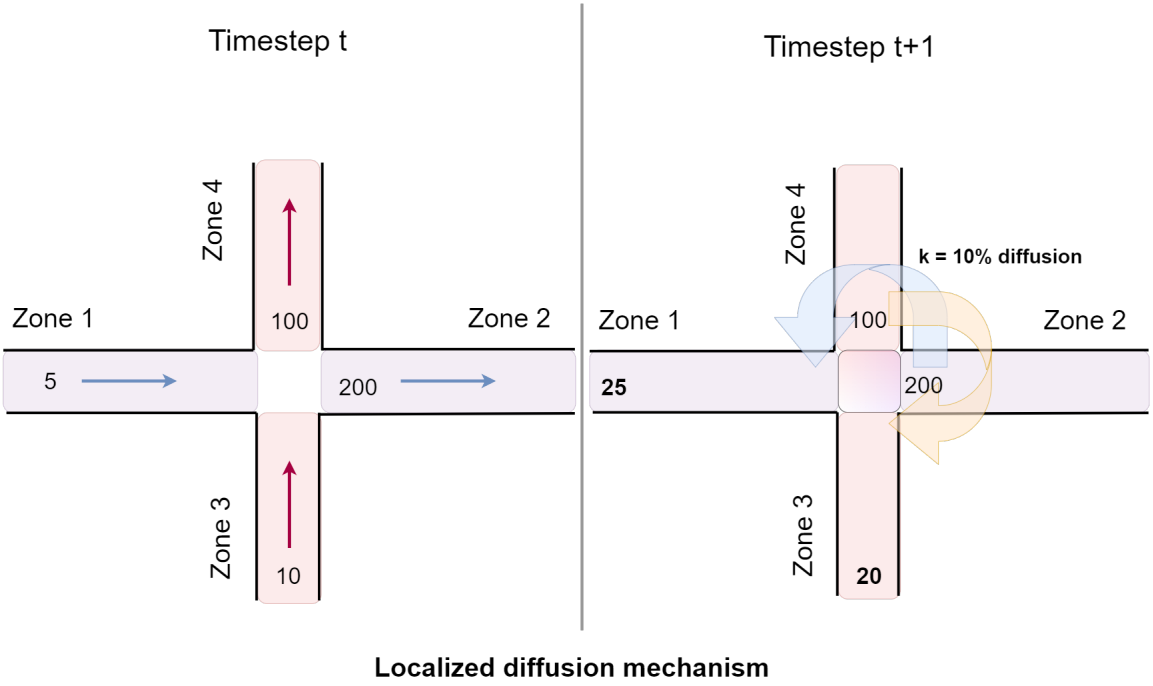

Figure 4.4: Localized Pheromone diffusion from anterior to posterior zone (eg. k = 0.1 or 10%)

The diffusion mechanism (Figure 4.4) is a supporting mechanism that entails the gradual spreading or dispersion of pheromone across a medium (environment). This diffusion process occurs as agents interact with their environment, releasing or absorbing substances that then propagate through the surroundings. Factors such as environmental conditions and agent actions influence the rate and extent of diffusion.

Similar to evaporation, diffusion plays a pivotal role in shaping the dynamics of our learning agents. It facilitates the distribution of information, by cascading [189] a local interaction into a system-wide influence. Specifically, in our pheromone-based approach, diffusion leads to the transfer of pheromone concentration from one zone to another, with closer zones receiving a larger share of concentration compared to those farther away. This concept can be visualized by considering each intersection as a miniature diffusive unit, highlighting how localized diffusions contribute to broader global diffusions across intersections (see Figure 4.5).

In our interconnected intersections scenario, where multiple intersections are linked, a new insight into the relationship between incoming and outgoing legs emerges. This

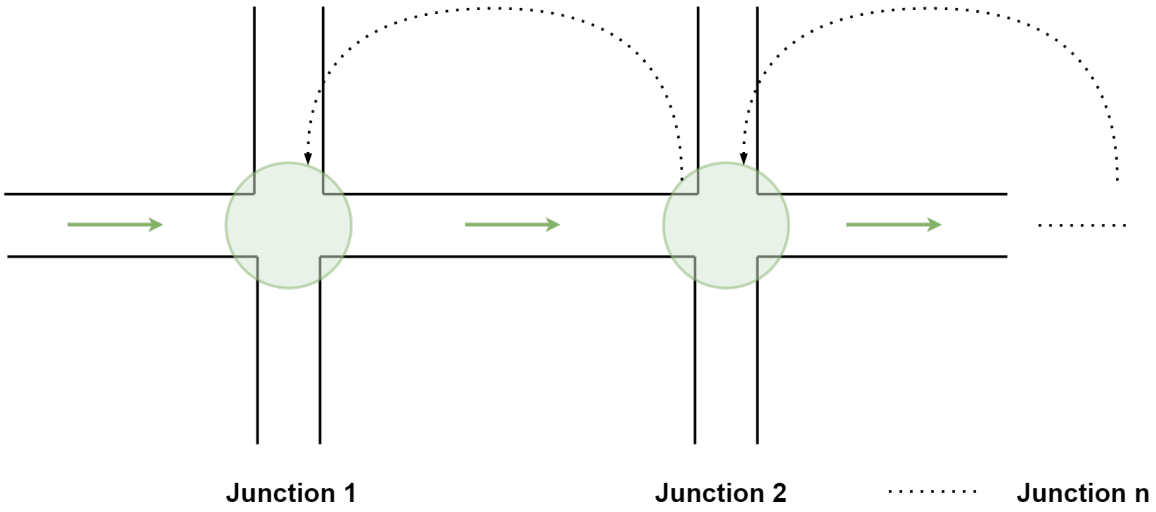

Figure 4.5: Pheromone diffusion between junctions for global feedback signal

observation, previously unattainable in single-intersection scenarios, helped make a design decision. In single intersections, agents on outgoing legs remain unaffected by pheromones from other legs. However, in interconnected intersection scenarios with multiple junctions, outgoing legs also serve as incoming legs for adjoining intersections. Consequently, agents entering any intersection are influenced by activity across all four legs.

Fine-tuning agent behavior hinges on adjusting the diffusion rate, which governs the dispersion of local pheromone values. By precisely defining the rate and direction of diffusion, we exert control over the spatial and temporal propagation of pheromone signals within the environment, balancing local exploration with global information dissemination. This diffusion process is regulated by a parameter called the diffusion factor (k), spanning inclusively from 0 to 1.

4.2.7 Lane-changing

In a highway scenario, vehicle agents have a restricted set of actions, which include lane change or maintaining their current lane position. To illustrate how pheromone signals facilitate indirect communication between agents, we delve into a straightforward example as seen in Figure 4.6.

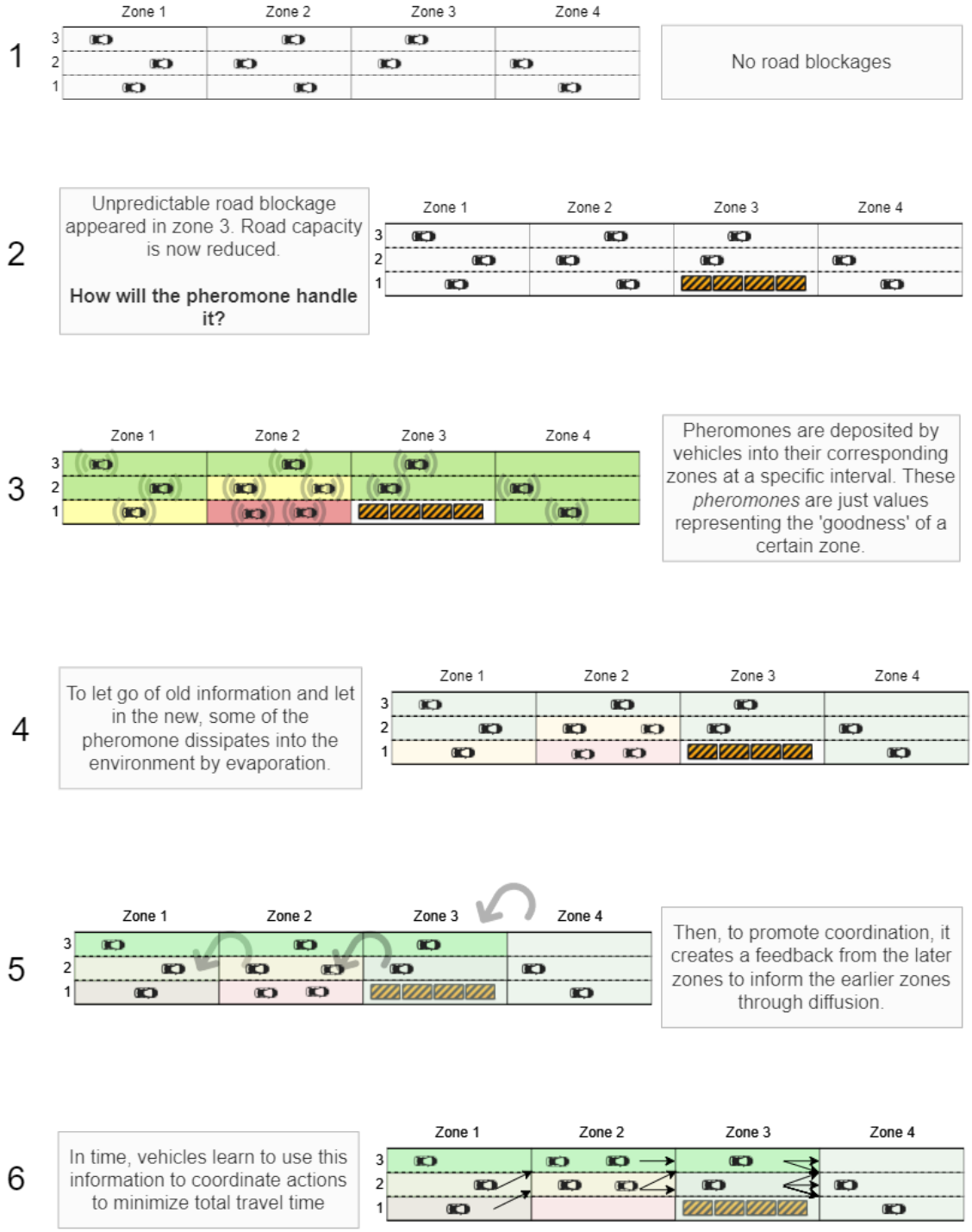

Figure 4.6: Highway pheromone process

Initially, the highway experiences no road blockages, and traffic flows smoothly across all lanes. Suddenly, an unpredictable road blockage materializes in zone 3, leading to a reduction in road capacity and potential traffic congestion. This road blockage could simulate anything from construction work to a vehicle accident. The vehicle agents deposit pheromones into their respective zones at regular intervals. These pheromone values represent the perceived *goodness* of each zone based on factors like traffic flow and lane availability, which could characterize congestion level. Over time, old pheromone information dissipates into the environment through evaporation, allowing room for new information to be incorporated. To foster coordination among vehicles, a feedback mechanism is established, enabling information from later zones to be communicated back to earlier zones through diffusion. As vehicles continually deposit and sense pheromone signals, they learn to coordinate their actions effectively to minimize overall travel time and navigate around road blockages. Vehicles navigate through different zones of the highway, depositing and sensing pheromone signals as they move. As road conditions change, such as the occurrence of a road blockage, the pheromone dynamics adapt accordingly, guiding vehicles' decision-making processes to optimize traffic flow and minimize congestion.

4.2.8 Intersection management

In an intersection scenario, we consider agents to either yield or take right-of-way at the intersection. To illustrate how pheromone signals facilitate indirect communication between agents, let's delve into the example as seen in Figure 4.7.

Let's consider an unsignalized junction. Initially, traffic flow at the intersection is evenly distributed, with equal levels of pheromone signaling from all incoming roads converging at the junction. As vehicle demand intensifies at one of the incoming zones (zone 1), pheromone signals indicate its reduced desirability. Consequently, the distribution of deposited pheromones alters between entering and exiting zones, establishing a pheromone gradient. This differential combined with our reward for equalization of local pheromone 4.2.4, should push vehicle agents to autonomously redistribute pheromone concentrations. Over time, the dissipation of old pheromone data occurs through evaporation, creating room for the integration of new insights. To enhance coordination, diffusion transpires from incoming

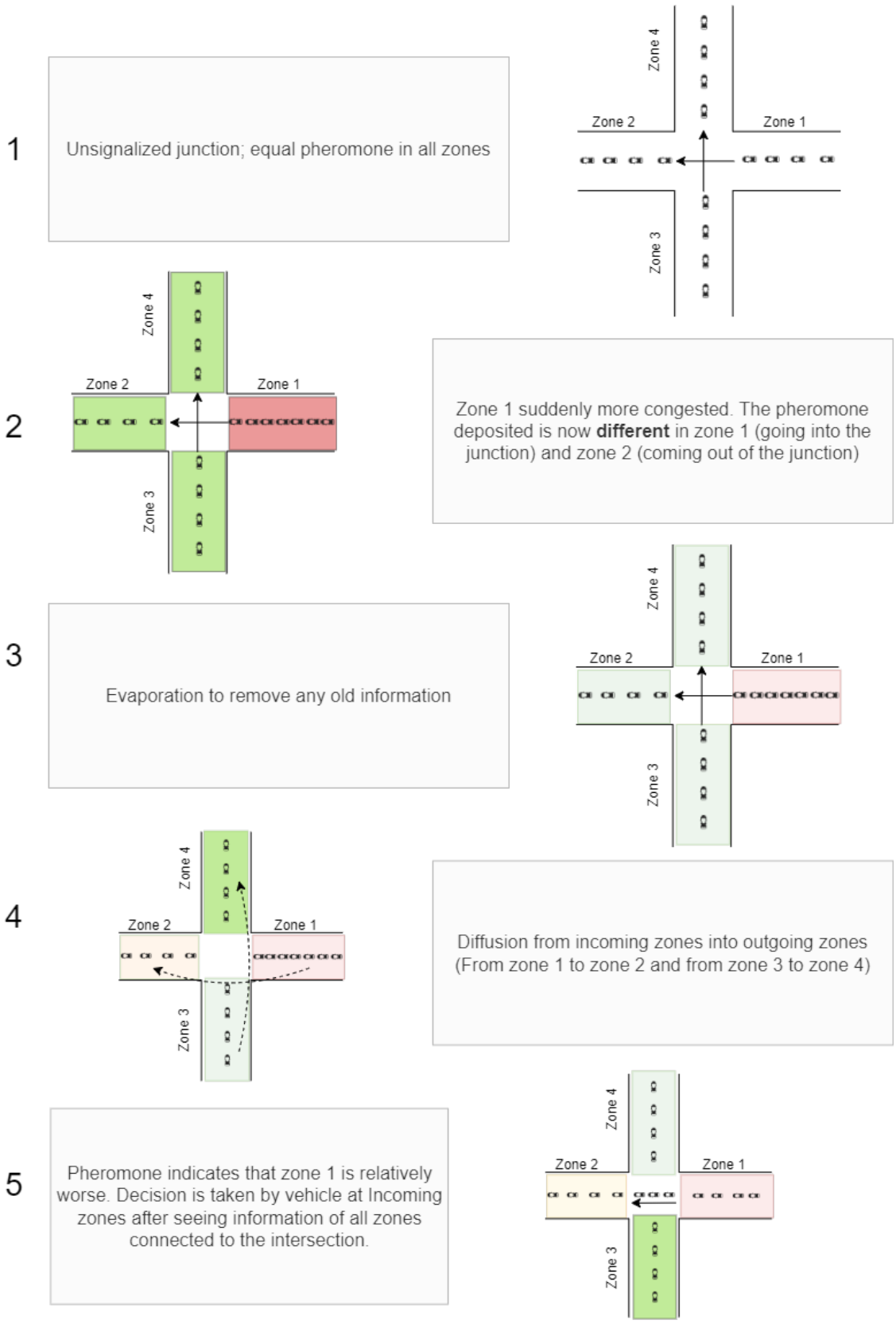

Figure 4.7: Junction pheromone process

to outgoing zones, conveying the heightened pressure or demand from incoming zones to their respective outgoing counterparts. Specifically, pheromones diffuse from zone 1 to zone 2 and from zone 3 to zone 4. Decision-making authority resides exclusively with vehicles in incoming zones, such as zone 1 and zone 3. These vehicles evaluate the pheromone values across all incoming zones, including their own, to determine whether to maintain their position or proceed through the junction.

A similar process occurs in interconnected intersections, however the diffusion process there is a distinct since the outgoing leg of a local intersection may be the incoming leg of another connected intersection. When focusing on a single intersection within a network of interconnected intersections, let's consider a specific incoming leg of the junction. Depending on the diffusion factor (k), a portion of the pheromone originating from the zones associated with the neighboring (or conflicting) incoming and outgoing legs, as well as the outgoing leg connected to the incoming leg, is diffused into the zone of this particular incoming leg. This process takes place across all 8 out of the 12 legs of the intersection that are regarded as incoming legs to certain intersections.

4.3 Integration of MARL with pheromones

Table 4.1: Pheromone Integration Methods

Method	Result	Simulation Behavior
1	Exploding gradients	-
2	Exploding gradients	-
3	Learning stable	Random
3 + 4	Learning stable	Visual pattern noticed

Key: 1 = Directly in Output Actions, 2 = Indirectly via Exploration Parameters/Noise, 3 = Directly in State Representation, 4 = Directly in Reward Function

Integrating pheromone signals directly into the output actions (Method 1) led to exploding gradients. This occurred because even minor fluctuations in pheromone levels resulted in disproportionately large changes in the selected actions. Since the policy network directly maps states to actions, these sudden variations caused unstable updates during training, leading to uncontrolled gradient magnitudes and divergence in learning.

Similarly, modifying exploration parameters or adding noise based on pheromone signals (Method 2) also resulted in exploding gradients. This method introduced a feedback loop where exploration behavior fluctuated excessively based on the external pheromone signal. If the signal varied too aggressively, the induced noise caused extreme policy updates, disrupting convergence and preventing stable learning.

Given these behaviors, it is logically concluded that any combination involving Methods 1 or 2 (direct action modification or pheromone-based exploration noise) would consistently lead to unstable or diverging learning behavior. The fundamental issue with these approaches was their direct influence on action selection, which caused excessive sensitivity to pheromone fluctuations and resulted in volatile gradient updates. Consequently, combinations such as (1 + 2), (1 + 3), (1 + 4), (2 + 3), and (2 + 4) would inherit these unstable properties, leading to either gradient explosions or vanishing updates. Extending this reasoning further, combinations involving three or more of these methods, such as (1 + 2 + 3), (1 + 2 + 4), (2 + 3 + 4), and (1 + 2 + 3 + 4) would compound these issues, making stable learning unlikely.

In contrast, integrating pheromone signals directly into the state representation (Method

3) maintained stable learning. By treating pheromones as part of the environment's state, the network could process them through multiple layers, mitigating abrupt changes in policy updates. This structured incorporation allowed for more controlled learning dynamics. However, upon visual inspection in SUMO, the vehicle agent behavior remained random when pheromone information was provided in the state alone. This suggests that merely supplying pheromone information did not guarantee its effective use unless the agent had a mechanism to interpret and leverage it within the learning process.

To address this, incorporating pheromone signals into the reward function (Method 4) created a feedback loop that reinforced the agentâs ability to utilize the pheromone information. Since the reward function shaped long-term behavior rather than immediate actions, this approach could avoid extreme gradient updates. The gradual reinforcement will allow the policy to adapt without destabilizing the training process. The combination of Methods 3 and 4 emerged as the only viable solution, where pheromone information in the state representation can be used through pheromone-dependent reward signals for feedback.

4.3.1 Directly with output actions

A policy network generates a vector or list of action probabilities. We attempt to integrate pheromone information with this vector as a form of additional noise associated with the action. To elaborate, the policy network's output of actions led to adjustments in neural network weights, though the actual actions were altered by the pheromone noise. These mixed messages destabilized the learning process.

An alternative approach was to modify the exploration parameters or noise introduced during action selection to incorporate pheromone cues indirectly. By influencing the exploration parameters based on pheromone signals, agents could adapt their exploration strategies dynamically in response to changing environmental conditions. We explore this in the next section.

4.3.2 Indirectly with exploration parameters/noise

Integration with the action selection mechanism involves modulating exploration noise parameters based on calculated pheromone contribution of the vehicle agent and can change the

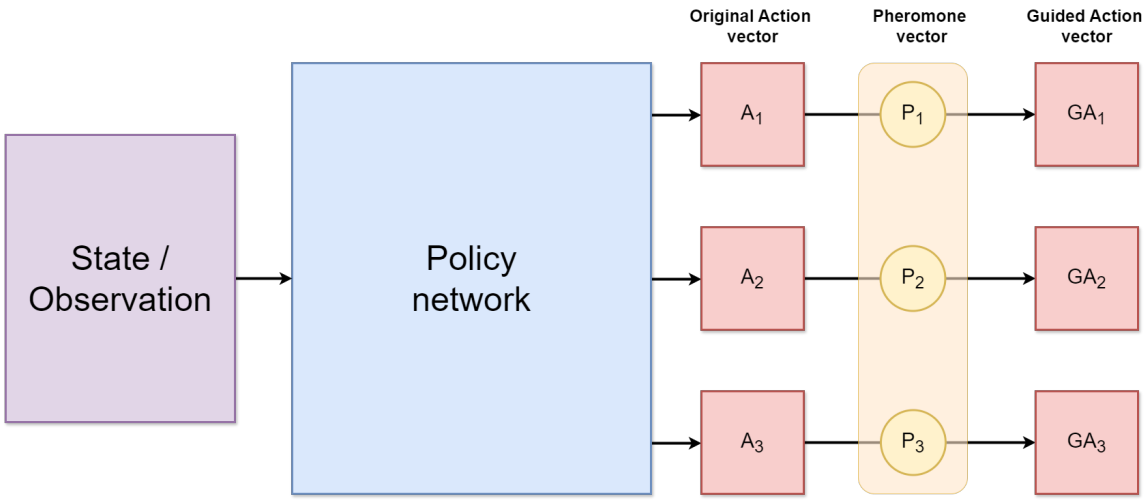

Figure 4.8: Pheromone guided action by directly combining with output action vector from the policy network

output of the action from the policy network. So exploration noise based on the calculated pheromone contribution of each vehicle agent would *guide* action selection. In practice, however, this approach often introduced instability into the training process, hindering overall learning performance. As a result, alternative strategies were explored to leverage pheromone information more effectively.

4.3.3 Directly with state

Integration with the state proves to be the most stable way to incorporate the information contained in the digital pheromones with the existing state in our MARL system. The pheromone operates differently from other state variables, as it continuously updates the state of the relevant part of the system. Unlike appending full state information from previous time steps to understand the history of state changes, the pheromone mechanism is simpler and more direct. Furthermore, the two primary processes of evaporation and diffusion facilitate the removal of outdated information and the dissemination of information across the system, fostering collaboration from local to global scales.

Incorporating pheromone information into the state representation within the Multi-Agent Reinforcement Learning (MARL) framework is a critical aspect of enabling agents to perceive and respond to environmental cues effectively. Unlike traditional approaches that rely solely on historical state information, the inclusion of pheromone data offers a dynamic,

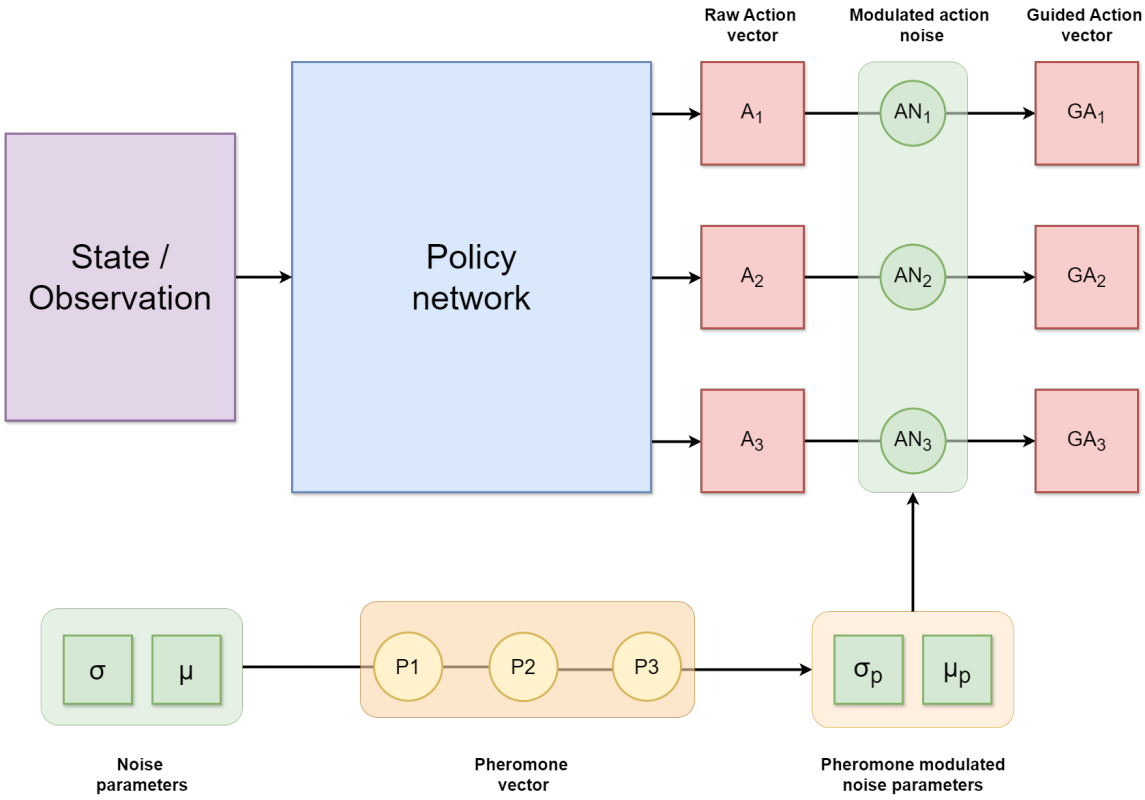

Figure 4.9: Pheromone guided action by influencing exploration noise or parameters using pheromone vector to affect the output action vector from the policy network indirectly

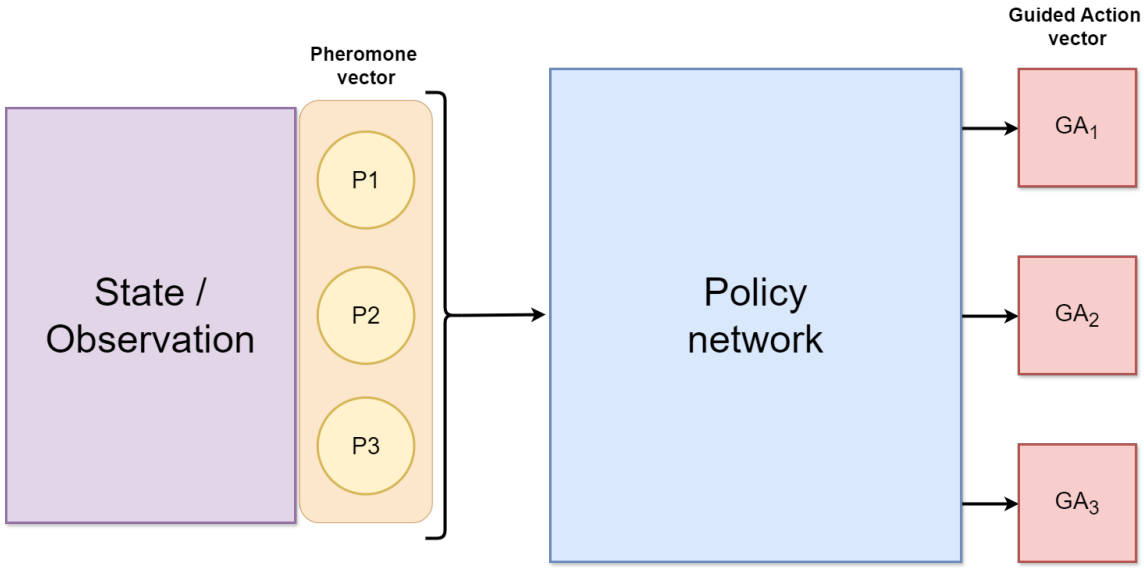

Figure 4.10: Pheromone guided action by simply appending the pheromone vector as part of the state

real-time perspective on environmental conditions and agent interactions, leading to more informed and context-aware decision-making.

The integration of pheromone information into the state representation begins with the collection and synthesis of pheromone data from the simulation environment. As agents move and interact within the environment, they deposit and sense pheromones, leading to changes in the distribution and concentration of pheromone signals over time. By incorporating this pheromone data into the state representation used by agents within the MARL framework, agents gain access to valuable insights into their surroundings, enabling them to make more informed and adaptive decisions.

Moreover, the integration of pheromone information into the state representation enables agents to perceive and respond to environmental cues dynamically. As pheromone signals evolve over time in response to agent activities and interactions, agents can adjust their behavior accordingly, leading to more responsive and adaptive behavior within the simulation environment. The refined state representation empowers agents to coordinate their actions in response to real-time environmental changes, thereby enhancing coordination towards system-wide objectives.

Furthermore, the integration of pheromone information into the state representation enables agents to leverage the processes of evaporation and diffusion to refine and disseminate pheromone information effectively. Through this integration, agents can effectively self-organize and adapt to their surroundings, fostering dynamic behavior within the simulation environment.

In summary, the integration of pheromone information into the state representation within the MARL framework is a critical aspect of enabling agents to perceive and respond to environmental cues effectively. By incorporating pheromone data into the state representation, agents gain access to valuable insights into their surroundings, leading to more informed and adaptive decision-making. Through this integration, agents can effectively coordinate their activities and optimize their performance based on real-time environmental conditions, leading to more efficient and adaptive behavior overall.

4.3.4 Directly in rewards

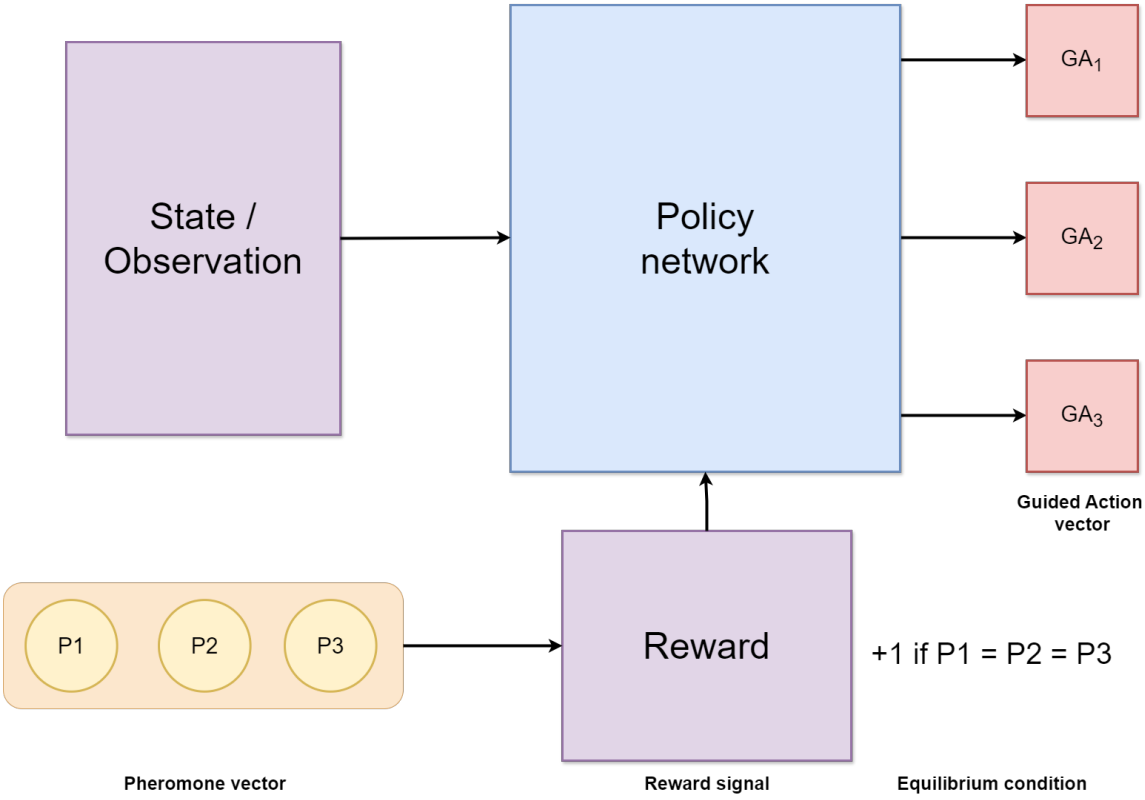

Figure 4.11: Pheromone guided action by rewarding balancing equilibrium of pheromones values

Integration with rewards involves engineering rewards for certain patterns of pheromones that can be seen as desirable. This sets a *stable* equilibrium state for the system to strive for. After evaluation (section 6.2), it proved to work well in conjunction with appending it to the state.

The integration of pheromone information into reward structures within the MARL framework represents a strategic approach to shaping agent behavior through targeted incentivization. By receiving positive feedback for desirable distributions of pheromone levels, agents can self-organize their actions towards achieving those local pheromone levels. Through evaporation and diffusion, these local interactions induce an emergent behaviour through self-organization, that redistributes traffic. It has been shown that redistributing traffic can potentially optimize traffic flow [54]. It creates a situation where the performance of the system is optimized as a whole, and no individual agent can find another action that can improve the system any more from that point [53]. The pheromone signals emitted by

CAV agents in zones are meant to guide the actions of fellow agents. We design a feedback signal that rewards the equalization of pheromones in zones that are locally perceivable by the agent. This does three things:

- The establishment of an objective for agents to equalize local pheromone levels promotes self-organization within the system.
- Through processes such as evaporation and diffusion, local interactions between agents result in emergent global dynamics, fostering a collective objective of achieving pheromone equilibrium across the system.
- Individual agents autonomously self-organize to optimize pheromone distribution, leading to the redistribution of traffic and consequent improvement in traffic flow efficiency.

In our approach, the reward system aims to stabilize pheromone levels towards equilibrium. Thus, a positive reward is assigned when the pheromone levels reach a state of equilibrium or balance. In a highway scenario, this translates to rewarding uniform pheromone levels across all lanes, while in an intersection scenario, it signifies equal pheromone levels across all legs of any intersection in the environment.

4.4 Induction of self-organization through digital pheromones

To seamlessly integrate our pheromone signals with the MARL algorithm, we opt to directly include them as part of our state representation. Additionally, we augment our approach by introducing rewards modeled after pheromone dynamics, enhancing the integration of the signal across multiple dimensions. The integration of pheromone within a generic MARL algorithm (Figure 4.12) is elaborated on in the following algorithmic description.

The Multi-Agent Pheromone-Enhanced Reinforcement Learning (PERL) algorithm (see 1) is designed to facilitate self-organization leading to emergent coordination for multiple agents within a partially observable non-stationary environment. At each time step, agents sense the pheromone levels in the environment and incorporate this information into their observations of the current state. Using a policy network, agents determine their actions, receiving an reward based on how close the perceived pheromone is to equalization. Rewards

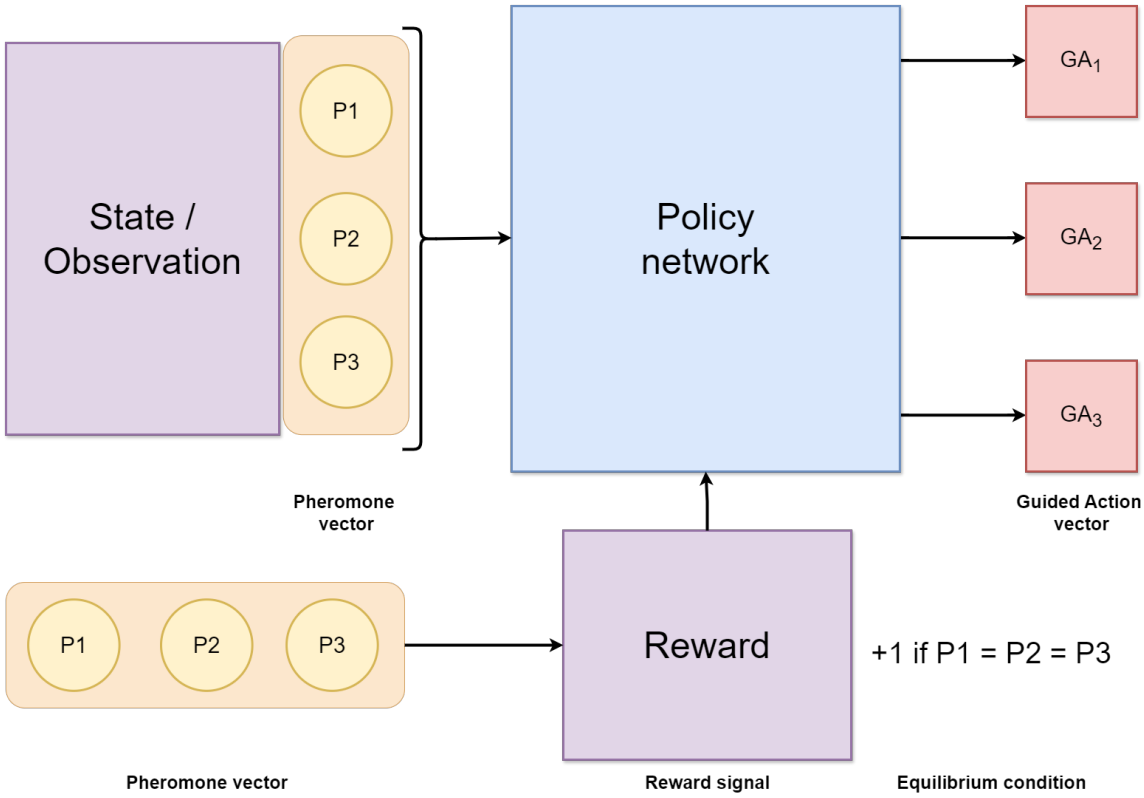

Figure 4.12: Pheromone guided self-organization

and next states are computed, and policy weights are updated accordingly. Additionally, the algorithm periodically updates the pheromone field based on the agents' contributions, adjusting for evaporation and diffusion effects to propagate information across the environment. By integrating MARL with indirect pheromone-based communication, PERL enables agents to collectively optimize their behavior towards achieving system-wide objectives.

At the beginning of this chapter, we discussed the essential components required for the self-organizing system. We have successfully accounted for positive and negative feedback mechanisms stemming from pheromone concentrations, capturing the notion of level of congestion. We have also addressed the amplification of fluctuations through pheromone deposition, evaporation, and diffusion processes, alongside acknowledging multiple interactions arising from the involvement of numerous agents/entities within the system. Thus, theoretically, our approach is poised to induce self-organization, yielding desirable emergent behaviors guiding by the reward for local equalization 4.2.4 of pheromone levels.

Algorithm 1 Multi-Agent Pheromone-Enhanced Reinforcement Learning (PERL)

Initialize policy network π
Initialize pheromone field p_f
for each step t **do**
    for each agent i **do**
        Agent i senses pheromone p in the environment
        Observe state s_i in the environment and append p to s_i for enhanced state E_i
        Apply action a_i using policy network π
        Reward $r \leftarrow r + e^{-\beta \mathbf{MAD}(p)}$, where β is the MAD multiplier (4.2.4)
        Compute reward r and next state s'
        Update policy weights π
    $s \leftarrow s'$
    if $t \mod pheromone\_update\_frequency$ **then**
        for each agent i **do**
            $x_i \leftarrow Features(agent_i)$
            $x_i \leftarrow Standardize(x_i)$
            $x_i \leftarrow MinMaxNormalize(x_i)$
            $contrib_i \leftarrow \sum_{i=1}^{n} x_i$
        Mean contribution $\mu_i \leftarrow \frac{1}{n} \sum_{i=1}^{n} contrib_i$
        Update with mean contribution and evaporate old values $p_f \leftarrow (1-j) * p_f + j * \mu$**, where j is the evaporation factor**
        Diffuse values to surroundings $p_{affected} \leftarrow (1-k) * p_{affected} + k * p_{influencer}$**, where k is the diffusion factor, p is pheromone in an influencing/affected zone**

4.5 Assumptions and Scope

The collision avoidance of our approach is handled by rule-based safety constraints. Since including it would make this a multi-objective reinforcement learning (MORL) problem, which can introduce additional complexity and challenges to the learning process [190]:

- Trade-offs between Objectives: When optimizing multiple objectives simultaneously, there often exists a trade-off between them. In our traffic scenarios, an agent might aim to minimize both travel time and collisions. However, actions that minimize travel time might increase collisions and vice versa. Balancing these trade-offs can make learning more challenging or unstable.
- Non-Convexity: The presence of multiple objectives often leads to non-convex optimization landscapes, where the objective function has multiple local optima. So our MARL algorithm may convergence to suboptimal solutions or oscillate between

different policies. For example, a vehicle may face uncertainty about the relative importance of different objectives, making it challenging to learn a stable policy.

There are works that address these challenges [191], however they also claim that the solution is computationally expensive [192]. So collision avoidance is not discussed in this thesis as it is out of scope. Additionally, PERL relies on emergent behaviour from self-organization of its agents and cannot guarantee consensus of agents in all situations. Lastly, the practical implementation of decentralized pheromones falls beyond the scope of this research as well; however, there is potential to use Vehicle-2-Vehicle (V2V) communication technologies like Vehicle Ad Hoc Network (VANET) technology [59, 60]. Digital pheromones can be stored in both, a centralized or decentralized manner [193].

5 Implementation and simulation environment

This chapter presents the implementation of agents and the simulation environment across all scenarios. We provide a comprehensive overview of the modules used to facilitate the development of pheromone-enhanced RL agents (PERLs) with integrated indirect communication using pheromones. The chapter begins by justifying the choice of simluation environments and algorithm frameworks used, followed by a description of the modules, agent implementations, their relationships, and any communication protocols employed, if any.

Specifically, we outline the key methods and functionalities implemented by the agents to enable pheromone-based communication. These methods are essential for the agents to interpret and respond to pheromone signals effectively. The simulation environment functions as a collection of sensors and actuators, facilitating the collection of data from the environment that agents require for decision-making. Additionally, we elucidate the interface between the agents and the simulation environment, which enables agents to affect their surroundings through the execution of actions. This chapter provides crucial insights into the technical aspects of the implementation process.

5.1 Simulation environment

Simulation of Urban MObility (SUMO) [104] is an open-source microscopic traffic simulation package designed to model and simulate traffic flows in urban environments. SUMO employs a microscopic simulation approach, wherein individual vehicles and pedestrians

are modeled and simulated in detail, allowing for accurate representation of traffic dynamics, interactions, and behaviors. The software supports a wide range of features, including vehicle routing, traffic signal control, lane changing, vehicle following models, and emission modeling. Additionally, SUMO provides an extensive set of APIs [194] and interfaces for integration with external tools and libraries, enabling advanced customization and analysis.

A lot of the research presented in Chapters 2 and 3 use SUMO as their simulation software for evaluation.

5.2 Algorithm Frameworks

In this section, we present the frameworks used for implementing customization to MARL in our proposed approach.

5.2.1 RLLib

RLlib [195] is an open-source reinforcement learning library developed by Ray, a distributed execution framework for Python that enables high-performance and scalable applications. It includes various state-of-the-art RL algorithms, such as DQN, PPO, and A3C, as well as support for custom algorithms. RLlib's architecture is less user-friendly and the documentation is more complex compared to other RL frameworks. As a result, navigating RLlib's features and understanding its inner workings requires a significant investment of time and effort. Because of this, we opt to use another framework for our evaluation.

5.2.2 StableBaselines3 and PettingZoo

StableBaselines3 (SB3) [196] is another open-source reinforcement learning library built on top of PyTorch. It is designed to provide a stable and easy-to-use implementation of state-of-the-art reinforcement learning algorithms. It provides a straightforward interface for training and testing RL algorithms.

PettingZoo (PZ) [197] is an open-source software library that provides a collection of diverse and customizable environments for studying multi-agent interactions. Since PZ environments are compatible with SB3, we opt to use PZ along with SB3.

5.3 Design requirements

When designing the simulation and implementation framework for our research, it's crucial to ensure that the environment is suitable for conducting experiments and validating the hypotheses effectively. Our design considerations:

1. **Scalability**: The implementation framework should be scalable to accommodate a large number of agents or entities.
2. **Flexibility**: The implementation framework should be flexible and configurable, allowing us to easily modify parameters, settings, and rules to simulate different scenarios and conditions. For example, using .YAML (Yet Another Markup Language) [198] files.
3. **Modularity**: The implementation framework must be modular, allowing different components or modules to be easily integrated, modified, or replaced.
4. **Interoperability**: We must ensure that our implementation framework can interact seamlessly with other tools and libraries such as SUMO, SB3, PZ and PyTorch.
5. **Observability and Monitoring**: The implementation framework must have mechanisms for monitoring and logging the state of the environment, including behaviour of agents, performance metrics and any data relevant for analysis.
6. **Reproducibility**: The implementation framework must have mechanisms (e.g., random seeds) that enable reproducibility of results.
7. **Performance**: Implementation must consider performance of the local system on which training and evaluation is to be performed. Performance can be improved using parallelization or distributed computing.
8. **Documentation and Support**: The implementation must have comprehensive documentation and versioning control, including function docstrings, clear comments, and versioning. Proper documentation and versioning are essential for understanding and maintaining the implementation, or even re-purposing it for a different application.

5.4 Modules

The class diagram (Figure 5.1) summarizes the structure of the implementation for our Pheromone-Enhanced RL (PERL) approach.

5.4.1 `SUMOConnector`

`SUMOConnector` is a base class that defines all the internal SUMO functions to connect, and control the simulation externally. The `SUMOSim` expands on it's functionalities.

5.4.2 `SUMOSim`

The `SUMOSim` class serves as a wrapper for interacting with the SUMO simulation. It inherits functionalities from the `SUMOConnector` class and extends them to provide specific simulation initialization and metrics computation.

Simulation Initialization

The class overrides the `_initialize_simulation()` method to define specific simulation initialization steps. It subscribes to various simulation constants and vehicle types, allowing for real-time monitoring of simulation parameters such as colliding vehicles and vehicle attributes like speed and acceleration, which we use for pheromone calculation (as specified in Chapter 4).

Metrics Initialization

Similarly, the `_initialize_metrics()` method is overridden to initialize relevant metrics for monitoring the simulation. For example, average number of lane changes per lane, or per vehicle may be captured so the result of a training or evaluation run may be plotted or analysed later. It maintains a list of episode rewards to track the performance of the simulation over time.

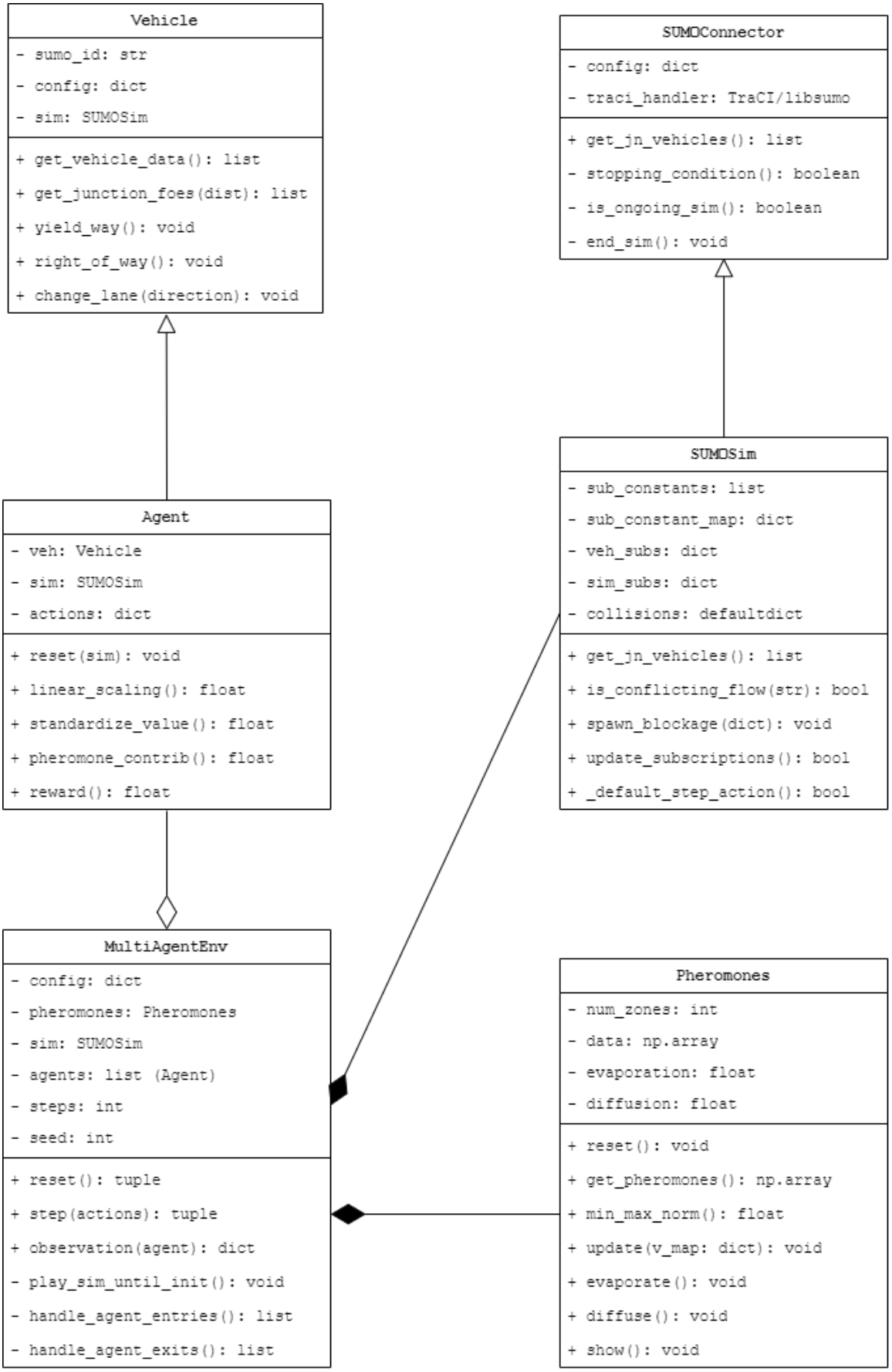

Figure 5.1: PERL class diagram

Simulation Steps

The `step()` method executes simulation steps until a condition is met. It interacts with the SUMO simulation environment by calling the `simulationStep()` method and performing default step actions specific to the simulation.

Additional Functionality

Additional methods such as `get_vehicles_at_intersection()` and `spawn_blockage()` provide functionalities for querying vehicle information and simulating blockages within the environment. These methods contribute to the flexibility and extensibility of the simulation framework.

The class also includes methods for logging and formatting subscription results obtained from the SUMO simulation. This allows for comprehensive monitoring and analysis of simulation data, aiding in debugging and performance evaluation.

The `SUMOConnector` class serves as a connector for interacting with the SUMO (Simulation of Urban MObility) simulation environment. It provides functionalities for initializing the simulation, executing simulation steps, and ending simulation. It also initializes the TraCI or libsumo APIs which help control actions in the environment later on.

Overall, the `SUMOSim` class encapsulates the functionality required to interact with the SUMO simulation environment, providing a robust and customizable framework for our traffic simulation.

5.4.3 MultiAgentEnv

This is practically the same class, but with slight variations depending on what type of scenario is being run in SUMO:

- `MultiAgentJunction`
- `MultiAgentHighway`

These classes implement a parallelized version of our environment for MARL using Petting Zoo (PZ). It manages the interactions between multiple agents (vehicles) and SUMO,

with the primary purpose of defining components of the RL cycle: `step`, `reset`, `action`, `reward`

Initialization

Upon initialization, the class sets up the environment parameters and initializes pheromones to guide agent actions. It configures the SUMO simulation environment and initializes internal variables to keep track of simulation steps and episodes.

Simulation Control

The class includes methods for resetting the simulation environment (`reset()`) and advancing simulation steps (`step()`). It handles agent entries and exits, updates pheromone information, and computes rewards based on agent actions and environment state.

Observation and Action Spaces

Methods like `observation_space()` and `action_space()` define the observation and action spaces for the agents in the environment.

Table 5.1: Observation and Action Spaces for MultiAgentHighway and MultiAgentJunction

Component	**MultiAgentHighway**	**MultiAgentJunction**
Observation Space	Pheromones Locations (one-hot encoded matrix) Agent Velocities	Pheromones At Intersection (boolean) Decided to Yield (boolean) Has Foes (boolean) Is Stopped (boolean) Foe Yielded (boolean)
Action Space	Vector of size 3 (left, right, stay)	Vector of size 2 (yield, right of way)

The reason `MultiAgentHighway` has a simpler observation space is because the vehicle agents did not have to take actions to control their speed since that was controlled directly by SUMO's rule-based Krauss model [199].

Agent Interactions

The class manages interactions between agents and the SUMO simulation environment, including handling agent entries and exits, updating agent states, and computing rewards based on agent actions. It provides functionalities for observing the environment, selecting actions with masking, and receiving rewards with masking.

Null and Illegal Actions

In our MARL framework, the challenge of addressing null and illegal actions arises when agents encounter situations where they are unable to execute the actions prescribed by their policy due to logical inconsistencies. Null actions occur in instances where the policy predicts actions for agents that are either non-existent or inactive within the environment. This happens when a vehicle has not yet been inserted into the simulation or when the vehicle has already left the simulation, but is in the list of active agents that PZ requires to identify all agents that will be part of an episode at the beginning of the episode.

On the other hand, illegal actions encompass actions that attempt to break the predefined rules or constraints of the environment. An example of this would be attempting to transition to the right lane when a vehicle is already occupying the rightmost lane, violating the traffic regulations.

To address these challenges, we employ action masking techniques. Null action masking ensures that inactive or non-existent agents do not take any actions, thereby preventing them from influencing the environment. On the other hand, illegal action masking restricts agents from taking actions that violate the environment's rules or constraints, maintaining the integrity of the simulation.

By implementing these masking strategies, we can mitigate the instability in learning caused by allowing illegal actions to be part of the training loop.

Helper data structures and functions

Various helper functions are implemented to facilitate environment management, agent interactions, and simulation control.

A useful data structure common in all our scenarios is the `vehicles_map` dictionary of sets. It holds the global information mapping each zone of the pheromone field with the set of vehicle agents that are in that zone in any timestep. We use this data structure to make pheromone aggregation (deposition) updates to our zones. The sets are useful since no duplicate can exist in these structures.

There are also helper functions that handle tasks such as creating and destroying agents, updating the `vehicles_map` at every step, and resetting the simulation environment.

5.4.4 `Vehicle`

The `Vehicle` class encapsulates functionalities and attributes pertinent to individual vehicles within the simulation environment. It serves as a crucial component for managing vehicle behavior and interactions.

Initialization

Upon instantiation, the class initializes various parameters such as vehicle identification (ID as in SUMO), simulation context, and configuration settings.

Speed Mode Configuration

Method `set_speed_mode()` configures the speed mode for the vehicle based on specified parameters, ensuring adherence to predefined speed constraints and intersection regulations. This functionality primarily serves to grant complete control over the vehicle's speed, ensuring that SUMO does not intervene in the speed management process.

Action Execution

Methods such as `yield_way()` and `right_of_way()` implement the execution of specific actions by the vehicle, such as yielding or asserting right-of-way at intersections in the SUMO simulation.

Since there is no built-in SUMO function to perform the yield or right-of-way actions, we write helper functions to mask illegal actions with reward penalization for situations where:

- the vehicle selected is in the middle of crossing an intersection when it decides to yield (and stop).
- the vehicles from conflicting flows both yield or assert right of way simultaneously.

Information Retrieval

The class provides functions to retrieve essential vehicle information, including position, speed, acceleration, and lane details.

Lane Change Control

Functionality for changing lanes (`change_lane()`) and monitoring lane change decisions (`get_lane_change_decision()`) is implemented to facilitate lane navigation and decision-making.

Visualization

The `highlight_vehicle()` method allows for visual highlighting of vehicles within the simulation environment, aiding in the visualization and analysis of vehicle behavior.

The `Vehicle` class plays a pivotal role in enabling the simulation and interaction of vehicles within the MARL framework, facilitating the execution of actions, retrieval of information, and visualization of vehicle behavior for analysis and decision-making.

5.4.5 `Agent`

Vehicle Initialization

The `JN_Agent` or `LC_Agent` class is responsible for managing the behavior of individual vehicles in the simulation environment. Upon initialization, an instance of `Vehicle` is created to represent the vehicle being controlled by the agent. Additionally, the available actions for the agent are defined based on the provided configuration.

Resetting the Agent

The `reset` method is invoked to reset the agent's state at the beginning of each simulation episode. This method subscribes the vehicle to relevant simulation variables, ensuring that it receives updated information from the environment.

Action Selection

The `act` method is responsible for selecting an action for the agent based on the current state of the environment. This method computes the desired action using a combination of the provided action space and environmental observations. Additionally, the `apply_action_masked` method is employed to ensure that the selected action adheres to the constraints imposed by the simulation environment.

Reward Computation

The `reward` function computes the reward for every step that an agent takes in the environment. Since we evaluate on multiple scenarios, we have varied reward functions for each scenario as described in Chapter 6. However, some commonalities can be observed.

- There is always a penalty for an illegal action, which is also always masked in the environment. The masking is a process by which we execute a neutral or no-action in the environment if the action selected by the agent's policy is impossible to execute. For example, an agent that is attempting to change lane to the right when it is already present at the rightmost lane. In this case, we penalize the agent as though it has a performed a "bad" action in a state, but continue the simulation by executing a null action or stay action for the vehicle representing the agent.
- Our proposed reward is calculated as $e^{-\beta \text{MAD}}$ (see Chapter 4), where MAD is the mean absolute deviation of local pheromone distribution as observed by the agent. This is also a common part of the reward in all scenarios.
- Another mostly common part of the reward is the "existence-is-pain" or "floor-is-lava" penalization of the agent, which can help mitigate the spare nature of the rewards

[200] in some cases, i.e., penalizing an agent at every step because it has not completed it's task or route can help with sparse rewards.

Observation Generation

The `observe` method generates observations of the environment for the agent. These observations include information such as the vehicle's current speed, acceleration, distance traveled, and the presence of obstacles or junctions in the vicinity. The observations are structured as an ordered dictionary to facilitate further processing by the agent.

Agent Destruction

The `destroy` method is provided for any necessary cleanup or resource deallocation required when the agent is no longer active.

5.4.6 `Pheromones`

The `Pheromones` class manages the pheromone field within the simulation environment. It offers methods for initializing, updating, and visualizing the pheromone distribution. Key functionalities include:

- `__init__()`: Initializes the pheromone field with parameters such as evaporation rate, diffusion rate, and initial values.
- `reset()`: Resets the values present in the pheromone field to either default values or random values within specified ranges (at episode reset).
- `update()`: Updates the pheromone field based on contributions from vehicle agents, implementing processes for evaporation and diffusion.
- `show()`: Displays the current state of the pheromone field for inspection.

The class also provides utility methods for min-max normalization of pheromone values to ensure consistency across different ranges.

6 Evaluation

In this chapter, we present an evaluation of PERL as a technique for cultivating coordination between MARL agents from local interactions in partially observable and non-stationary or dynamic environments. We outline the evaluation's objectives and the metrics employed to gauge PERL's performance. Additionally, we elucidate the experimental scenarios and parameters utilized in this evaluation. Finally, we report and analyse the results of the evaluation.

6.1 Evaluation Objectives

The goal of the evaluation of PERL presented in this chapter is to establish how well the stigmergic mechanism integrated with our MARL agents can contribute to learning a policy that leads to an emergent state of equilibrium by self-organizing in a partially observable non-stationary environment. The primary objective behind designing PERL was to offer a decentralized, self-organizing, multi-agent optimization technique that enhances system performance by fostering global coordination through local stigmergic interactions.

In Chapter 4, we outlined how the design of PERL meets these criteria. However, the significance of PERL as a technique for enhancing MARL hinges on its performance across various evaluation scenarios.

In the first scenario, we simulate multi-CAV navigation through dynamic obstacles or blockages on a multilane highway, further contributing to the non-stationarity of the environment. Subsequently, we simulate multi-CAV negotiation through a single 4-way intersection with two adjacent conflicting unidirectional unilane flows, and four interconnected 4-way intersections, each presenting two adjacent conflicting unidirectional unilane flows. All these

scenarios exhibit non-stationarity due to the non-linear interactions between multiple agents in the system, along with partial observability as CAV agents can only observe their local vicinity.

PERL can be deemed successful if it fulfills the following performance criteria:

1. PERL demonstrates superior performance compared to rule-based methods, showcasing its ability to surpass simple heuristic strategies.
2. PERL exhibits enhanced performance compared to MARL approaches that lack stigmergic mechanisms, highlighting the efficacy of integrating stigmergy for achieving better coordination among agents in a partially observable, non-stationary environment.
3. The emergent behaviour arising from PERL balances pheromone levels across the system and therefore redistributes and equalizes traffic flow on a system level.
4. PERL demonstrates its effectiveness by improving the performance of system policies across various environmental conditions, such as dynamic blockages. Moreover, it proves adaptable to different policy characteristics, whether off-policy or on-policy, showcasing its versatility.

In the next section we present the evaluation baselines we use to assess the performance of PERL.

6.2 Baselines

In this section, we describe the baselines against which we evaluate PERL. As part of the evaluation, we will deploy the simulator's [104] built-in heuristic policies, custom heuristic policies, and Independent RL (IRL or MARL without communication).

6.2.1 SUMO's built-in heuristic policy

Depending on the type of scenario, we use two distinct heuristic policies from SUMO. In scenarios where CAV agents must change lanes to traverse through a multilane highway with obstacles, we use the SL2015 lane-changing model [201]. In scenarios where CAV agents on unilane roads must negotiate at intersections, the CAVs are controlled by a set of

junction model parameters [202]. In both these types of scenarios, the Krauss car-following model [199] contributes to controlling longitudinal actions of CAVs like acceleration and deceleration.

Lane-changing scenarios

The SL2015 lane-changing model [201] controls the lane-changing behaviour and lateral dynamics of vehicles. CAVs are driven by a combination of rules and parameters [203] designed to mimic real-world driving behavior:

1. **Gap Acceptance:** CAVs assess the availability of suitable gaps in adjacent lanes before initiating a lane change, aiming to find a gap that allows for smooth merging without causing disruptions to the flow of traffic.
2. **Speed Adjustment:** CAVs may adjust their speed to create or utilize gaps in traffic, ensuring safe and efficient lane changes.
3. **Speed Gain:** CAVs may choose to change to a specific lane for the reason that they estimate they can have a higher speed on that lane.
4. **Safety Considerations:** The SL2015 model prioritizes safety by incorporating factors such as lane occupancy, lane curvature, and the presence of obstacles. Vehicles are programmed to prioritize safe maneuvers, avoiding risky lane changes that could lead to accidents.
5. **Traffic Fluidity:** This involves balancing the need for vehicles to change lanes to reach their destinations with the overarching goal of maintaining overall traffic fluidity.

Junction negotiation scenarios

The negotiation at an intersection without traffic lights in SUMO is primarily governed by the following rules [202]:

1. **Right-of-Way:** CAVs approaching the intersection typically yield to vehicles already within the intersection or to vehicles approaching from the right.

2. **Gap Acceptance:** CAV wait for a sufficient gap in the conflicting traffic stream before proceeding at intersections
3. **Speed Adjustment:** CAVs may adjust their speed as they approach the intersection based on the presence of other vehicles and their own priority to ensure safe negotiation.
4. **Turning Behavior:** CAV negotiate turns at intersections according to predefined turning rules, such as giving way to vehicles going straight or vehicles in the opposing lane turning left.

These rules collectively govern how vehicles negotiate intersections in SUMO's junction models, even in the absence of traffic lights. The specific parameters and configurations for these rules can be adjusted in SUMO's configuration files to simulate different intersection behaviors and traffic scenarios, however we use the default parameters [202] for it.

6.2.2 Custom rule-based policies

Depending on the type of scenario, we use two distinct custom heuristic policies. In scenarios where CAV agents must change lanes to traverse through a multilane highway with obstacles, they use a predefined probabilistic rule-based policy based on local pheromones (see Chapter 4), and in scenarios where CAV agents on unilane roads must negotiate at intersections, they use a conditional rule-based policy based on the max pressure concept [120].

Lane-changing scenarios

As detailed in Chapter 4, CAV agents perceive pheromones within their current zone as well as the immediately adjacent lateral zones in adjacent lanes. In highway scenarios, we use a probabilistic rule-based policy. Action selection involves computing the softmax over locally perceived pheromone levels, followed by selecting the action with the highest probability for execution.

$$\text{action\_idx} = \text{argmax}_i \left(\frac{e^{p_i}}{\sum_{j=1}^{n} e^{p_j}} \right) \tag{6.1}$$

p_i : Perceived pheromone level associated with action i

p_j : Perceived pheromone level associated with action j

Junction negotiation scenarios

As discussed in Chapter 4, CAV agents detect pheromone levels on the legs of the local intersection they are navigating. In scenarios where agent actions are limited to yield and right-of-way actions, we employ and adapt the concept of max pressure [120] to select actions. Basically, this concept states that the incoming flow on the leg of the intersection with the highest pressure must be allowed to pass first, with pressure determined by the number of vehicles present on a leg. However, since our pheromone concentrations indicate the desirability of a location (see Chapter 4), we give right-of-way to the incoming flow of traffic on the leg with the lowest local pheromone concentration:

$$\text{Action} = \begin{cases} \text{Right of way (go)} & \text{if agent is on leg with least pheromone} \\ \text{Yield (stop)} & \text{otherwise} \end{cases} \tag{6.2}$$

6.2.3 IRL (MARL without communication)

Independent Reinforcement Learning (IRL) represents MARL that does not rely on communication mechanisms between agents. In all our evaluated scenarios, CAV agents will make decisions independently without influencing each other through indirect communication via pheromones. During our evaluation, we use PPO for lane-changing scenarios and TD3 for intersection negotiation scenarios (see chapter 2) as our base RL algorithms for IRL (and PERL).

In the next section we present the metrics we use to evaluate the performance of PERL.

6.3 Evaluation Metrics

To assess PERL's performance, we utilize a set of metrics tailored to the specific task for applications in traffic management:

Table 6.1: Evaluation metrics

Common	**Lane-changing**	**Junction negotiation**
Cumulative mean Gini index (ratio)	Mean recovery time after perturbations (s)	Mean waiting time (s)
Cumulative mean Shannon entropy ($bits$)	Mean number of lane-changes	Mean number of stops (halts)
Total simulation time (s)		
Mean travel time (s)		

We utilize the Gini index [204] (see section 6.3.1) to assess the uniformity of the distribution of pheromones or agents within the traffic environment. This metric reflects the emerging coordinated behavior achieved through self-organization, indicating the proximity of the system to equilibrium (see Chapter 4 for discussion on Nash equilibrium [53]). In addition to the Gini index, we support these investigations with Shannon entropy [205], which serves as a measure of the disorder or randomness within a system. Here, equilibrium corresponds to maximum entropy, as it signifies a system that has achieved balance and no longer exhibits spontaneous change tendencies. We also derive the rate of change of Gini index and Shannon entropy for pheromone and agent distribution in the system, helping us understand how fast the system is striving moving towards equilibrium. We go into detail about calculation in further sections (6.3.1 and 6.3.2).

Congestion metrics (see section 6.3.3) provide an insight into the congestion levels in traffic. For lane-changing scenarios, the mean recovery time (section 6.3.4) helps us see how well an approach adapts to additional non-stationary conditions like blockages. Additionally, excessive number of lane-changes (section 6.3.5) or halts (section 6.3.6) can create bottlenecks that affect traffic fluidity, making it an insightful metric.

6.3.1 Gini index

We introduce Gini index [204], a measure based on the Lorenz curve [206], used to quantify the degree of economic inequality within a population or distribution. A higher Gini index indicates greater income or wealth inequality, while a lower index suggests more equitable distribution.

The Gini index provides valuable insights into the distributional patterns of resources. Mathematically, the Gini index is calculated using the Lorenz curve , which plots the cumulative share of income or wealth against the cumulative share of the population. The Gini index (or ratio) is equal to the area between the Lorenz curve and the line of perfect equality, divided by the area under the line of perfect equality. In our evaluation, we calculate Gini index using the mean absolute deviation (MAD) [185]:

$$\text{MAD} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}| \tag{6.3}$$

Where: n is the number of data points

x_i is the value of the ith data point

$\bar{x}$ is the mean of the data points

So following from the formula in (6.3), we calculate Gini index [204] as:

$$\text{Gini Index}(G) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}} \tag{6.4}$$

Where: n is the number of data points

x_i and x_j are the values of the data points at positions i and j.

$\bar{x}$ is the mean of the data points.

$0 \leq G(X) \leq 1$

The Gini index (6.4) is always in the range 0 (perfect equality) and 1 (perfect inequality), inclusive. Since Gini index can only work with positive values, we normalize our pheromone values in the range 0 to 1 (from -1 to +1) [188].

6.3.2 Shannon entropy

Shannon entropy [205] quantifies the uncertainty or randomness in a distribution. In the context of evaluating resource distribution equality, it provides a quantitative measure of the dispersion or concentration of probability mass in a distribution, making it a useful tool [207] for assessing the equality of pheromone or CAV agent distribution (see Figure 6.1) in the environment. We also use it as an additional metric to further support the Gini index.

According to the second law of thermodynamics [208], a system at equilibrium has achieved maximum disorder or randomness. At this state, energy and matter are evenly spread throughout the system, eliminating gradients and potential for spontaneous change. In simpler terms, equilibrium represents the system's most disordered state, where all possible arrangements of its components have been explored, resulting in maximum entropy.

In the context of evaluation of approaches, a distribution with high Shannon entropy indicates that pheromone concentrations or CAV agents are evenly distributed across the environment. Conversely, a distribution with low entropy suggests that pheromone concentrations are concentrated in specific segments of the environment. Shannon entropy [205] is given by:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{6.5}$$

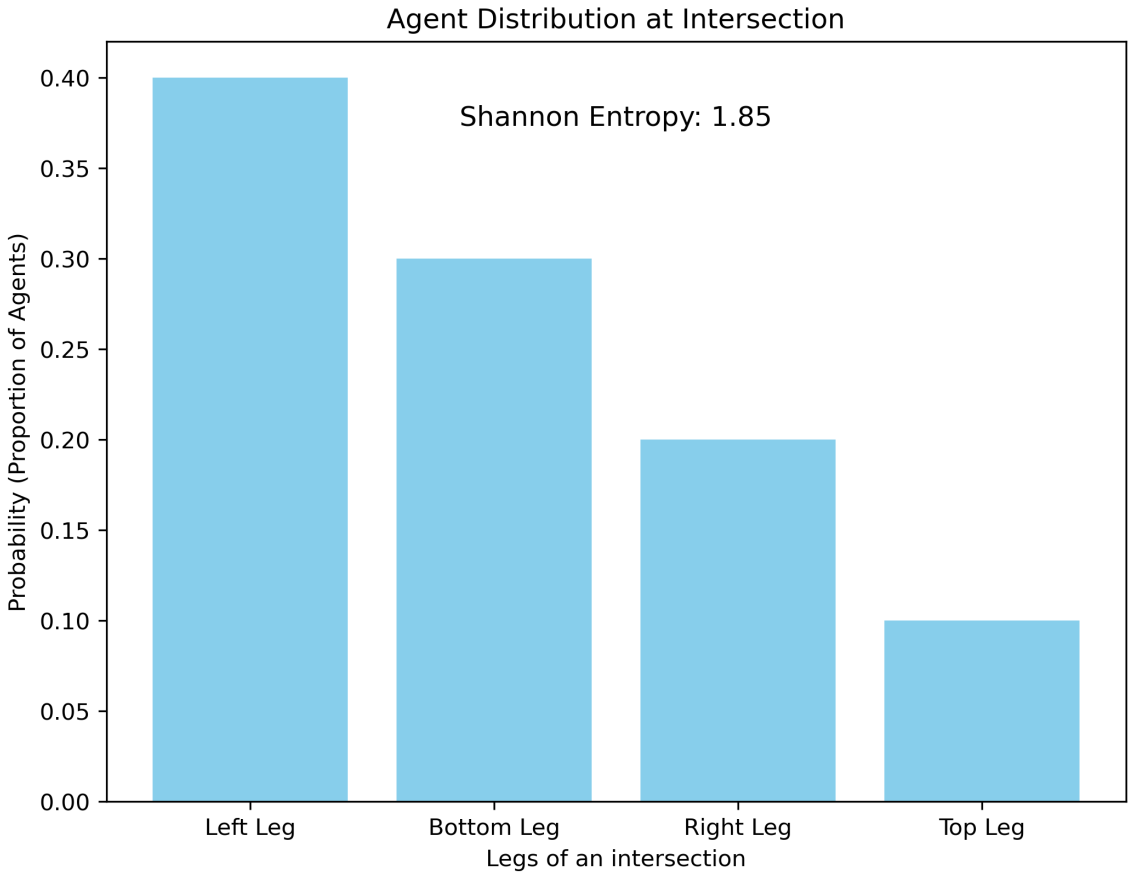

Figure 6.1: Shannon entropy to measure agent distribution

Where: $H(X)$ is the Shannon entropy of the system

p_i is the probability of event i

x is the total number of events or states in the system.

$0 \leq H(X) \leq \log_2(n)$

6.3.3 Congestion Metrics

Metrics like total simulation time, mean waiting time and mean travel time collectively provide an assessment of congestion levels within the traffic environment [25].

6.3.4 Mean recovery time

The goal of mean recovery time [209] is to evaluate the adaptability of CAV agents to dynamic disturbances like blockages in scenarios with lane-changing. Specifically, it measures the time difference between the moment a disturbance occurs, causing a perturbation in pheromone levels across lanes, and the point at which the distributed pheromone levels return to the similar Gini index values (± 0.1) observed before the disturbance.

6.3.5 Mean Number of Lane Changes

This metric quantifies the frequency of lane changes performed by CAV agents in scenarios with lane-changing. It provides insights into the level of traffic fluidity and stability, as excessive lane changes may contribute to congestion [210].

6.3.6 Mean Number of stops (or halts)

Halting and waiting at an intersection causes delay [211]. The mean number of stops by CAV agents provide insights into the level of traffic fluidity and congestion.

6.4 Evaluation Scenarios

In this section we describe the scenarios that we used to evaluate the suitability of PERL for partially observable (see section 2.9.2) non-stationary (see section 2.9.3) environments. Some parameters and details are consistent across all evaluations:

Simulation Termination

The simulation concludes when either all inserted CAV agents reach the end of their route or a maximum of 3600 simulation steps is reached.

Experimental Reproducibility

To ensure reproducibility, we assess the scenario by averaging the outcomes from 5 evaluation runs. Each evaluation run employs the parameters we will discuss further, but with different

random seeds.

In our simulations, the behavior of vehicles, including their route selection, starting positions, and destinations, is predefined. This means that we specify these parameters for each vehicle before the simulation begins, rather than allowing them to be determined dynamically during the simulation. By defining the behavior of vehicles in advance, we can control and manipulate specific scenarios within the simulated environment. This predefined behavior also allows for more reproducible and consistent experimentation, as the same scenarios can be recreated and analyzed repeatedly.

Varying parameters

For our evaluation, we conducted 10 trial runs, each with a different set of parameters, such as vehicle starting speed at insertion and the total number of vehicles in the system. This variability is why we report both the mean and standard deviation of the results.

In the lane-changing scenario, the number of vehicles is capped at 100, as this represents the maximum capacity of the selected road segment. Each vehicle is approximately 5 meters in length, and the simulation incorporates safety gaps and vehicle-following dynamics that create additional spacing. Increasing the number of vehicles beyond this limit would not alter system behavior, as excess vehicles would be unable to enter the simulation, leading to repeated behavioral patterns within a single run.

To ensure a representative range of traffic conditions, each trial varies both the total number of vehicles and their starting speeds, in addition to accounting for blockages. The maximum speed is capped at 20 m/s (72 km/h) to align with typical real-world driving conditions. Since vehicle starting speeds influence how quickly the road reaches saturation, they indirectly affect traffic density. By averaging results across 10 runs with varying parameters, we capture a spectrum of traffic densities, from sparse to dense. This approach provides a more comprehensive evaluation of PERLâs performance under diverse traffic conditions.

Hyperparameter choices

The hyperparameter values used in our experiments are primarily based on the default settings provided in the original papers for each corresponding algorithm. These default values have been established through prior research and extensive testing, ensuring a reasonable balance between performance and stability.

The selection of these algorithms was motivated by their relatively lower sensitivity to hyperparameter tuning, as discussed in a prior chapter. This property makes them well-suited for our study, as it reduces the need for extensive manual tuning and allows for a fair comparison across methods. While some minor adjustments were made to account for the specific characteristics of our traffic environment, such as scaling factors for reward signals and minor modifications to learning rates to improve convergence, the overall hyperparameter configurations remained consistent with their original implementations.

By relying on well-established defaults and selecting algorithms with lower hyperparameter sensitivity, we mitigate the risk of overfitting to specific scenarios and ensure that our results reflect generalizable learning behavior rather than fine-tuned optimizations.

We present the simulation environment and parameters, and then describe the specific details of each scenario used.

6.4.1 Scenario 1 : Unidirectional Multilane highway with dynamic blockages

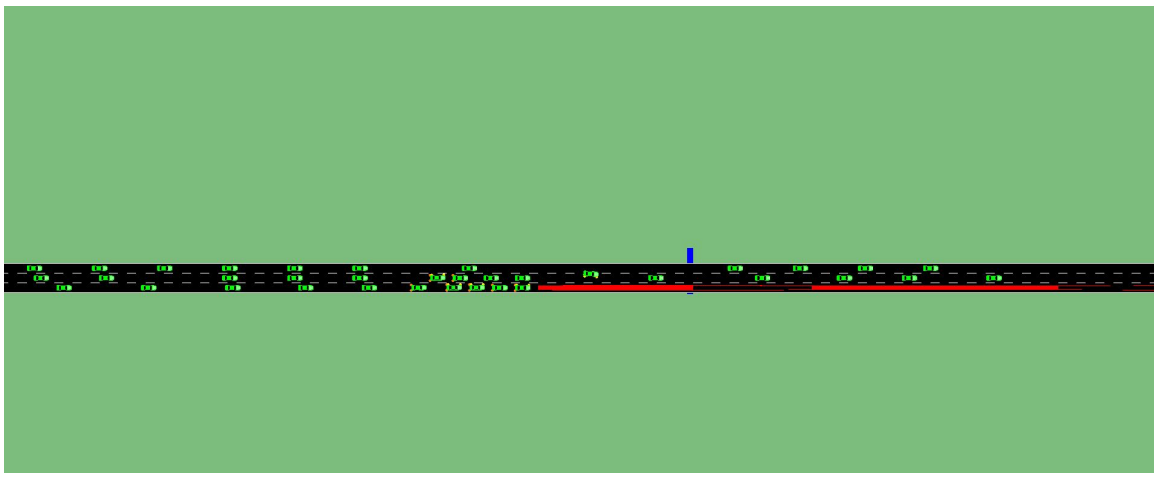

Figure 6.2: Learning lane-changing on a multilane highway

In this scenario, multiple CAV agents navigate through dynamic disturbances (blockages)

on a unidirectional multilane highway. The blockage location, start time, duration were randomly selected (based on random seed) for evaluation runs. The objective of the CAV agents is to get to the end of the straight road by learning to change lanes in non-stationary conditions. Scenario related parameters are set for various experiments:

Parameter	Values
Road Length (m)	1000, 1500
Number of Lanes	2, 3, 5
Maximum Number of agents	20, 100
Maximum possible speed (m/s)	20
Maximum possible acceleration / deceleration (m/s^2)	$\pm$ 4

Table 6.2: Scenario parameters for evaluation in highway scenario

State Space

The state space includes location of the ego CAV agent and pheromone concentrations in zones within the range of perception (see section 4.2.3). This means that the CAV agent can perceive pheromones in its current zone, and zones on its immediate left and right on adjacent lanes. Location is a one-hot encoding that helps agent know which lane it currently is on, so it can correspond it with the perceived pheromone values.

Table 6.3: State space for evaluation in highway scenario

Feature	**Unit**	**Size**	**Range**
Pheromones	Floating-point vector	1 x 3	[-1,1]
Locations	One-hot vector	1 x number of lanes	0 or 1

Pheromone calculation

The pheromone contribution (see section 4.2.5) from CAV agents is the mean of standardized and normalized (between -1 and +1) sums of velocity and acceleration of each agent in that zone. This is because mean speed and acceleration are among the factors [25] that represent traffic flow and characterize the congestion level of a zone.

Action Space

A CAV agent on a multilane highway road can execute one of 3 actions:

- Change lane to the left
- Change lane to the right
- Stay in the same lane

The speed of the CAV agent is controlled by SUMO [199], since we want to focus on learning effective lane-changing to navigate around blockages or similar dynamic disturbances.

Blockages and Obstacles

Blockages are virtual obstacles on the road with set parameters like size, location, and duration. CAV agents must learn to navigate around them or wait for clearance.

Table 6.4: Blockage Parameters for evaluation in highway scenario

Parameter	Values
Number of blockages	1
Size of blockages (m)	Randomly chosen from (200, 300)
Location in lane (m)	Randomly chosen from (750, 1000)
Blockage on lane (lane index)	Randomly determined at start of simulation
Duration of blockage (s)	Randomly chosen from (10, UNTIL END)

Since we train the system with a mixed blockage parameter scheme, where each episode during training has a randomized number, location, size and duration of the blockages, we also evaluate it in the same way.

Reward System

We use the mean absolute deviation (MAD) [185] of pheromone levels on zones within the CAV agent's perception (current zone and zones on immediately adjacent lanes to the left and right) to calculate a reward proportional to how equalized the distribution is. The closer the MAD is to 0, the closer the reward is to +1. CAV agents do not receive any

reward for reaching the end of the highway (since there are no learnable actions to go faster longitudinally), but they receive a constant negative penalty for every second that they have not made it to the end. This is to encourage them to accumulate lower number penalties by completing their route faster. The reward at every step for a CAV agent is designed as:

$$\text{Reward} = \begin{cases} e^{-\beta \text{MAD}} \text{ for every step} \\ -1 \text{ for every step that agent has not reached the end} \\ -0.1 \text{ for each intended illegal action} \end{cases} \tag{6.6}$$

β is a parameter that acts as the multiplier for equalizing locally perceived pheromone distribution. We set it to 1 for all our experiments. Masking of illegal actions (as discussed in 4) is the prevention of execution of such actions, overriding the actual behaviour with a masked action, while also penalizing the agent for intending to do so. In practice, we mask actions by executing a neutral action (no-action) in the environment.

Zones and Pheromone Field

The pheromone field is divided into zones, and CAV agents contribute to the pheromone field within the zone in which they are based. The evaporation and diffusion factor are both set to half their maximum value to balance out the processes of incorporation of new data (both from diffusion from other zones and aggregation within current zone) and removal of old information.

Table 6.5: Pheromone Parameters

Parameter	Values
MAD multiplier (β)	1
Number of zones	4
Evaporation Rate	0.5
Diffusion Rate	0.5
Pheromone update frequency (simulation steps)	1

Algorithm and Hyperparameters

For this particular scenario's evaluation, PPO (refer to section 2.8.3) is used as the primary RL algorithm over which we build our stigmergic mechanisms for training the CAV agents in our MARL system. Specific hyperparameters for PPO used during evaluation of highway scenario:

Table 6.6: PPO Hyperparameters

Hyperparameter	Value
SGD minibatch size	128
SGD iterations	30
Learning Rate (α)	5e-5
Train batch size	4000
PPO Clipping parameter	0.3
Discount factor (γ)	0.99

We run this algorithm for 1000000 training steps or until convergence based on no mean reward improvement in 5 evaluations done during training. The frequency of evaluation is every 100000 training steps.

6.4.2 Scenario 2 : Negotiation at single intersection

In this scenario, multiple CAV agents interact at an unsignalized intersection. The intersection consists of two incoming legs from the left and bottom directions, and two outgoing legs leading to the right and top directions. Each leg of the intersection is unilane and unidirectional, with a length of 100 meters. The goal of the CAV agents is to navigate through the junction and minimize the time taken for all agents to reach the end of the straight road. This task involves learning to coordinate among themselves at the intersection, which introduces non-stationary dynamics. Various experimental parameters are configured to conduct different experiments within this scenario.

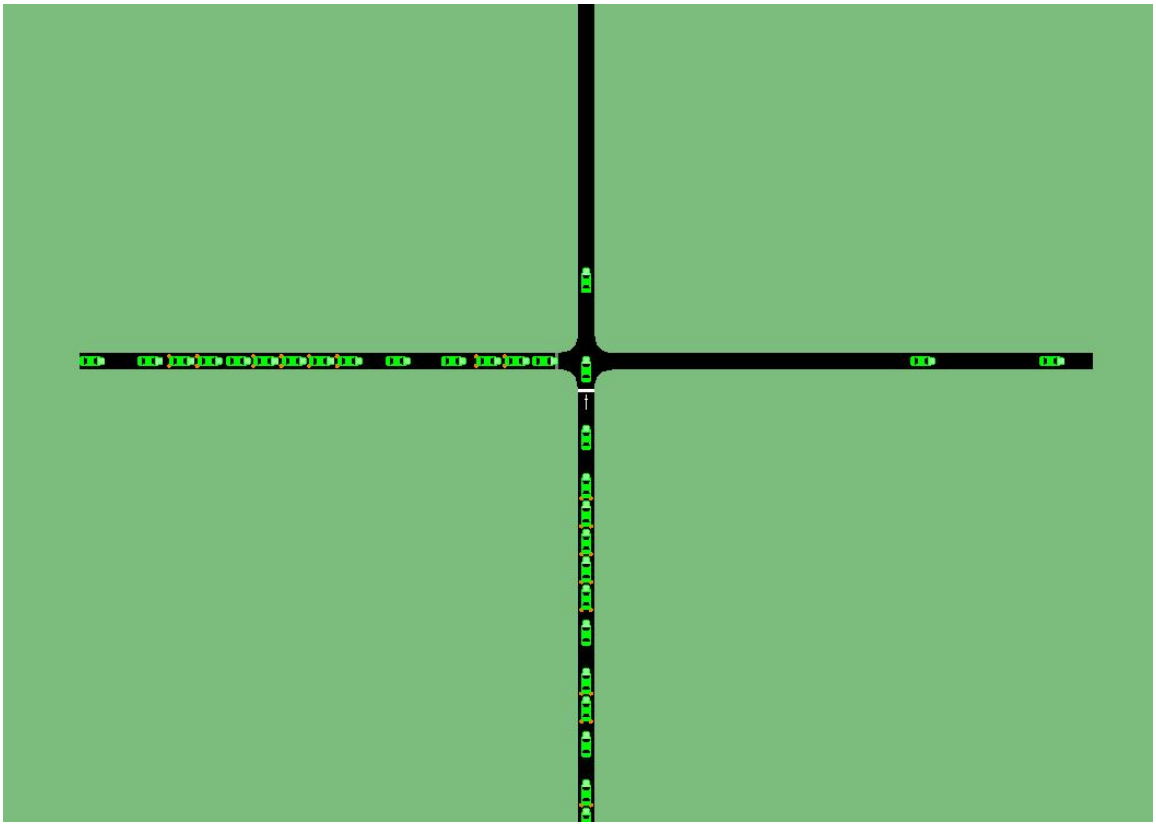

Figure 6.3: Learning negotiation at an intersection

Table 6.7: Scenario parameters for evaluation in single intersection scenario

Parameter	Values
Road Length (m)	100 for each leg
Number of Legs	4
Number of Lanes	1
Maximum Number of agents	100 on each leg
Maximum possible speed (m/s)	20
Maximum possible acceleration / deceleration (m/s^2)	± 4

State Space

The state space comprises the perceived pheromones from the agent's perspective, supplemented by a series of boolean checks aimed at fostering consensus among agents.

The boolean checks provide information on various aspects of the agent's situation. These checks determine if the agent is in the midst of crossing an intersection, has already opted to yield, encounters any foes within range in the conflicting direction, is currently at a standstill, or if the foe has yielded in the conflicting leg. Unlike the pheromones, which primarily influence agent behavior based on environmental cues, these additional

Table 6.8: State space for evaluation in single intersection scenario

Feature	Unit	Size	Range
Pheromones	Floating-point vector	1 x 4	[-1,1]
At Intersection	Boolean	1	True or False
Decided to Yield	Boolean	1	True or False
Has Foes	Boolean	1	True or False
Is Stopped	Boolean	1	True or False
Foe Yielded	Boolean	1	True or False

features serve to support the learning process by enabling agents to gauge consensus among themselves. This additional information supplements the pheromone signals, helping to enhance the overall coordination and decision-making of the agents.

Pheromone calculation

The pheromone contribution from CAV agents, outlined in Section 4.2.5, is determined as the average of standardized and normalized sums of velocity, acceleration, and waiting time observed within a particular zone or incoming leg of the intersection. These metrics, influenced by congestion levels [25], are intended as improved proxies for estimating the traffic pressure experienced by each leg of the junction [120].

Action Space

A CAV agent negotiating at a single intersection only has 2 actions:

- Yield (stop, and let the conflicting leg pass)
- Right-of-way (Go, and take right-of-way)

Reward System

Here, we also utilize the mean absolute deviation (MAD) [185] of pheromone levels within the perceptual range of CAV agents (including the current zone and adjacent lanes) to compute a reward that reflects the equality of pheromone distribution. A MAD closer to 0 corresponds to a reward closer to +1, indicating a more balanced distribution. In addition

to a small reward for reaching their destination, CAV agents incur a constant, minimal negative penalty for every second they spend without completing their route. This penalty incentivizes agents to minimize their overall penalty accumulation by completing their routes swiftly.

Furthermore, agents face penalties for being stationary (speed = 0), discouraging strategies that involve balancing positive rewards from pheromone equalization with minimal penalties from prolonged simulation participation. The reward at every step for a CAV agent is designed as:

$$\text{Reward} = \begin{cases} e^{-\beta \text{MAD}} \text{ for every step} \\ +5 \text{ for every step that agent reaches the end of their route.} \\ -0.5 \text{ for every step that agent is at rest} \\ -0.1 \text{ for every step that agent has not reached the end} \\ -0.5 \text{ for each intended illegal action} \end{cases} \tag{6.7}$$

The parameter β serves as a multiplier for equalizing locally perceived pheromone distributions, and in our experiments, we consistently set it to 1. We also explain masking of actions in the previous scenario description.

Zones and Pheromone Field

In our approach, the pheromone field is partitioned into zones, with CAV agents contributing to the pheromone field within their respective zones, each corresponding to a leg at the intersection in this case. To maintain consistency, we maintain the practice of setting the evaporation and diffusion factors at half of their maximum values just as before.

Algorithm and Hyperparameters

For this particular scenario's evaluation, TD3 (refer to section 2.8.3) is used as the primary RL algorithm over which we build our stigmergic mechanisms for training the CAV agents in

Table 6.9: Pheromone Parameters

Parameter	Values
MAD multiplier (β)	1
Number of zones (and legs)	4
Evaporation Rate	0.5
Diffusion Rate	0.5
Pheromone update frequency (simulation steps)	1

our MARL system. Specific hyperparameters for TD3 used during evaluation of intersection scenarios:

Table 6.10: TD3 Hyperparameters

Hyperparameter	Value
Replay buffer (experiences)	1000000
Learning starts (steps)	18000
Learning Rate (α)	1e-3
Minibatch size	512
Discount factor (γ)	0.99

We execute this algorithm for 1000000 training steps (consistent with PPO in lane-changing) or until convergence, determined by no mean reward improvement in 5 evaluations conducted during training. Evaluations occur every 1/10th of the total training steps, resulting in an evaluation frequency of every 100000 training steps.

6.4.3 Scenario 3 : Negotiation at multiple connected intersections

We expand the single intersection scenario to include four interconnected intersections, where multiple CAV agents negotiate conflicts at each junction. These junctions exhibit different directional conflicts, including flows from left to right, bottom to top, top to bottom, and right to left, creating distinct conflict scenarios at each intersection. Notably, turns are not considered in this setup, as the primary focus is on negotiating and resolving conflicts at the junctions.

Each leg of the intersection is unilane and unidirectional, spanning a length of 100 meters. The objective for CAV agents is to navigate through the interconnected intersections while minimizing the time taken for all agents to reach the end of the straight road. Achieving this goal requires learning to coordinate among themselves at multiple intersections, thereby

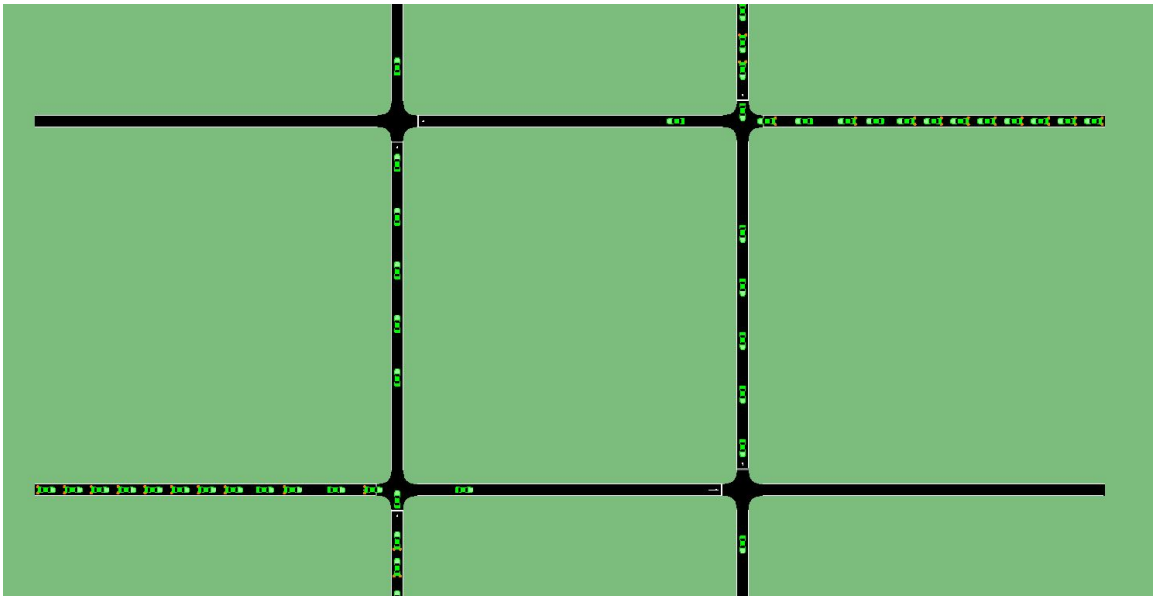

Figure 6.4: Learning negotiation at interconnected intersections

promoting non-stationary dynamics. Various experimental parameters are configured to conduct experiments within this scenario.

Table 6.11: Scenario parameters for evaluation in multiple interconnected intersections scenario

Parameter	Values
Road Length (m)	100 for each leg
Number of Legs	12
Number of Lanes	1
Maximum Number of agents	100 on each incoming leg of the system (4)
Maximum possible speed (m/s)	20
Maximum possible acceleration / deceleration (m/s^2)	± 4

State Space

In both the single intersection and multiple interconnected intersections scenarios, we utilize pheromone signals from each leg of every intersection, supplemented by additional information conducive to achieving consensus among agentsâdetails that are absent from the pheromone signals alone. This approach remains consistent across both scenarios. The perception range encompasses the legs of the intersection that the agent currently occupies as an incoming leg. As a result, the state space for multiple interconnected intersections closely resembles that of the single intersection scenario.

Table 6.12: State space for evaluation in multiple interconnected intersection scenario

Feature	Unit	Size	Range
Pheromones	Floating-point vector	1 x 4	[-1,1]
At Intersection	Boolean	1	True or False
Decided to Yield	Boolean	1	True or False
Has Foes	Boolean	1	True or False
Is Stopped	Boolean	1	True or False
Foe Yielded	Boolean	1	True or False

It's worth noting that the absence of consensus encoding in the pheromones is significant, as this multiple interconnected intersections necessitates a stronger emphasis on achieving consensus among agents due to increase in complexity of interactions. The included variables encompass boolean checks that provide insights into various aspects of the agent's situation. These checks ascertain whether the agent is currently traversing an intersection, has opted to yield, detects foes within range in the conflicting direction, remains at a standstill, or if the foe has yielded in the conflicting leg. While pheromones primarily influence agent behavior based on environmental cues, these additional features serve to augment the learning process by allowing agents to assess consensus among themselves. This supplementary information complements the pheromone signals, thereby enhancing overall coordination and decision-making among the agents.

Pheromone calculation

The pheromone calculation is done exactly the same way as it is done for the single intersection scenario. But the diffusion process is modified a bit (see Chapter 4). The pheromone calculation follows the same procedure as in the single intersection scenario. However, there are modifications to the diffusion process (detailed in Chapter 4). Unlike in the single intersection case, where values are diffused from the zone for an incoming leg to the corresponding outgoing leg, here, values from zones representing every other leg of the local intersection are diffused into the zones for incoming legs. It's important to note that out of the 12 legs in the scenario, 8 of them serve as incoming legs. This occurs because there are outgoing legs that also function as incoming legs for adjacent connected intersections. Essentially, these legs comprise all segments shared by two consecutive intersections.

Action Space

The action space is the same as it was in the single intersection version:

- Yield (stop, and let the conflicting leg pass)
- Right-of-way (Go, and take right-of-way)

Reward System

The rewards also remains unchanged from the single intersection scenario:

$$\text{Reward} = \begin{cases} e^{-\beta \text{MAD}} \text{ for every step} \\ +\,5 \text{ for every step that agent reaches the end of their route.} \\ -\,0.5 \text{ for every step that agent is at rest} \\ -\,0.1 \text{ for every step that agent has not reached the end} \\ -\,0.5 \text{ for each intended illegal action} \end{cases} \tag{6.8}$$

The parameter β serves as a multiplier for equalizing locally perceived pheromone distributions, and in our experiments, we consistently set it to 1. We also explain masking of actions in the previous scenario description.

Zones and Pheromone Field

In our methodology, we divide the pheromone field into zones, with each zone corresponding to a leg at the intersection. CAV agents contribute to the pheromone field within their respective zones. To ensure consistency, we continue the practice of setting the evaporation and diffusion factors at half of their maximum values, as in previous iterations. The pheromone update frequency refers to how often the operations of aggregation (deposition), evaporation, and diffusion are executed relative to the simulation time steps, which is set to 1 for all experiments.

Table 6.13: Pheromone Parameters

Parameter	Values
MAD multiplier (β)	1
Number of zones (and legs)	12
Evaporation Rate	0.5
Diffusion Rate	0.5
Pheromone update frequency (simulation steps)	1

Algorithm and Hyperparameters

Continuing the extension from the single agent intersection, we choose TD3 (refer to section 2.8.3) as the primary RL algorithm over which we build our stigmergic mechanisms for training the CAV agents in our MARL system. Specific hyperparameters for TD3 used during evaluation of single intersection scenario is the same as before.

Table 6.14: TD3 Hyperparameters

Hyperparameter	Value
Replay buffer (experiences)	5000000
Learning starts (steps)	180000
Learning Rate (α)	1e-3
Minibatch size	512
Discount factor (γ)	0.99

We execute this algorithm for 5000000 training steps or until convergence, determined by no mean reward improvement in 5 evaluations conducted during training. Evaluations occur every 1/10th of the total training steps, resulting in an evaluation frequency of every 100000 training steps.

6.5 Results and Analysis

In this section we analyze the performance of PERL with respect to the evaluation objectives outlined in Section 6.1. We evaluate the performance of PERL when compared to the performance of our baselines (section 6.2).

Table 6.15: Highlights of PERL vs. baselines for blockage on multilane highway

Metric	**SUMO**	**HP**	**PPO-IRL**	**PPO-PERL**
Simulation time (s)	192.5 ± 0.2	329.8 ± 1.6	274.2 ± 0.8	**183.2 ± 0.9**
Mean recovery time (s)	39.7 ± 1.76	119.97 ± 7.85 ±	85.91 ± 5.43	**33.19 ± 2.1**
Mean travel time (s)	33.15 ± 0.44	40.98 ± 0.28	38.77 ± 1.17	**32.59 ± 0.36**
Mean pheromone distribution Gini	0.67 ± 0.03	0.69 ± 0.02	0.67 ± 0.04	**0.59 ± 0.02**
Mean pheromone distribution entropy ($bits$)	1.54 ± 0.01	1.61 ± 0.02	1.57 ± 0.03	**1.64 ± 0.01**
Mean agent distribution Gini	0.58 ± 0.02	0.61 ± 0.02	0.63 ± 0.09	**0.56 ± 0.01**
Mean agent distribution entropy ($bits$)	1.86 ± 0.01	1.48 ± 0.03	0.97 ± 0.05	**1.92 ± 0.02**

6.5.1 Lane-changing on highways with dynamic disturbances

In this section, we compare chosen performance metrics for a multilane highway scenario, with upto 100 inserted agents. Blockage placements are determined at the start of the simulation, randomly from a list of choices as outlined in Chapter 4. For our evaluation, the randomly chosen predefined parameters set a blockage at between 450m - 750m on the 1000m length of the road, on the right most lane. The blockage persists until the end of the simulation. We have 4 zones which divides the road into 250m segments for representation in the pheromone field. PERL performs better than the baselines in terms of simulation time, mean recovery time and mean travel time. We investigate the distribution of pheromone levels on lanes (Figure 6.5) and it shows the cumulative mean of gini index of pheromone distribution over time during the evaluation. The distribution of pheromone seems to be closer to equilibrium in the case of PERL (refer to Table 6.15)

Results (Table 6.15) also indicate that PERL performs better than SUMO, in terms of simulation time and mean travel time, by $\sim$ 5% and $\sim$ 2%, respectively. Another interesting result is the mean recovery time, which shows that PERL adapts to blockages placed dynamically, $\sim$ 18% faster (Figure 6.8) than the second best (SUMO). The entropy (Figure 6.6) also supports the idea of equalization of pheromone levels being good for the overall performance of the CAV agent. The higher the entropy, the closer the system is to exploring all possible organizations of its entities (explained in section 6.3.2). The system is moving towards a situation in which no agent has an incentive to unilaterally deviate from

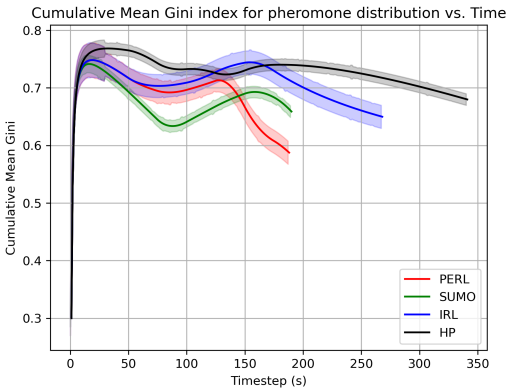

Figure 6.5: Gini index for pheromone distribution

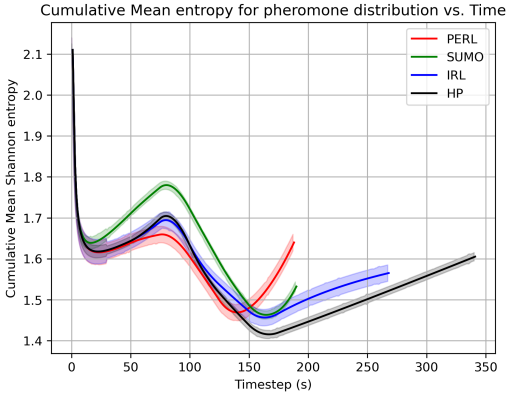

Figure 6.6: Shannon Entropy for pheromone distribution

their chosen strategy, given the strategies chosen by the other agents. This follows from the concept of Nash equilibrium [53], where no individual agent has a tendency to change the actions they have been executing at equilibrium. It also supports the results for Gini index of pheromone and agent distribution.

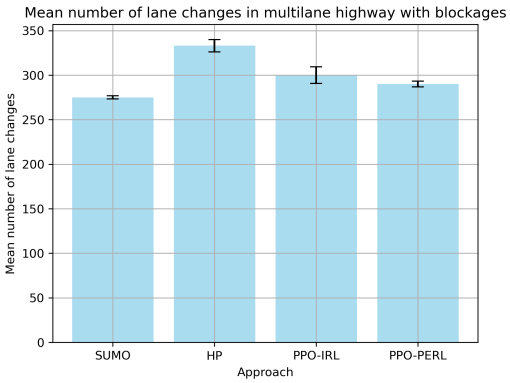

Figure 6.7: Mean number of lane changes in multilane highway with blockages

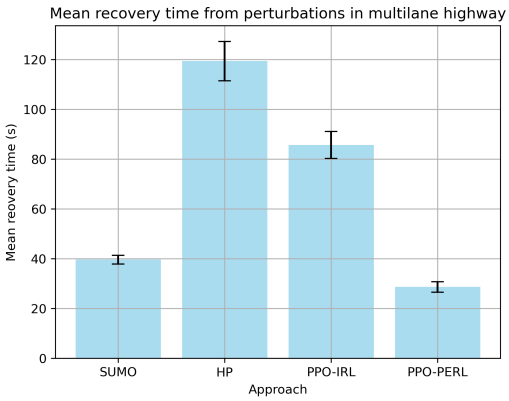

Figure 6.8: Mean recovery time in multilane highway with blockages

Additionally, the mean number of lane changes (Figure 6.7) tells us that the Heuristic policy (HP) we used performed the worst with the excessive lane-changing maneuvers. The reason for this phenomenon lies in the frequency of lane-changing actions performed per second within the simulation environment. Additionally, the occurrence of every agent selecting the same action is more likely in the case of the Heuristic Policy (HP). This results to an instantaneous alteration in pheromone levels as a result of a widespread reorganization of vehicle behavior. This leads to a cycle of non-stationarity in states and actions. Similarly, PERL also exhibits comparable behavior in specific edge cases, particularly when the initial

pheromone values are set to 0 instead of being randomly distributed. In such cases, the absence of initial pheromone variations can lead to a more synchronized agent response, where multiple vehicles select similar actions simultaneously. This synchronization may cause abrupt shifts in pheromone levels across the environment, triggering cascading effects on lane-changing behavior. The resulting fluctuations contribute to the overall instability observed in traffic patterns, particularly during the early stages of learning, before the policy has sufficiently adapted to diverse traffic conditions. While PERL demonstrates improved adaptability over time, these edge cases highlight the sensitivity of pheromone initialization and its influence on emergent agent behaviors.

Furthermore, the mean recovery time (Figure 6.8) shows that PERL is $\sim 18\%$ better than the next best performing baseline (SUMO). As described in Section 6.3, the recovery time denotes the duration from the moment a CAV agent encounters a disruption (such as a blockage) to the onset of a notable decline in the Gini index of the global pheromone distribution, signifying disturbance. This duration extends until the Gini index of the distribution returns to its original value or diminishes further, signifying stabilization or redistribution of traffic.

Figure 6.8 indicates that PERL exhibits a relatively high standard deviation, suggesting variability in its recovery consistency compared to the SUMO simulator. This discrepancy is likely attributable to the discrete nature of action selection in MARL, where agents make decisions at fixed intervals (once per second) rather than continuously adjusting their behavior in real time. This discretization may introduce fluctuations in recovery efficiency, as agents lack the ability to make fine-grained corrections between decision steps. In contrast, the SUMO simulator, which relies on predefined traffic dynamics, demonstrates more stable and predictable recovery patterns.

6.5.2 Negotiation at single intersection

In this section, we compare chosen performance metrics for a single junction scenario, with upto 100 agents each for the 2 incoming legs in the intersection, i.e., 200 agents. Figure 6.9 shows the cumulative mean of gini index of pheromone distribution over time during evaluation. SUMO performs the worst in terms of pheromone distribution over the legs of

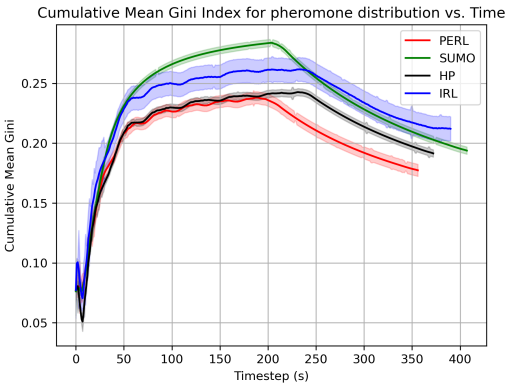

Figure 6.9: Gini index for pheromone distribution

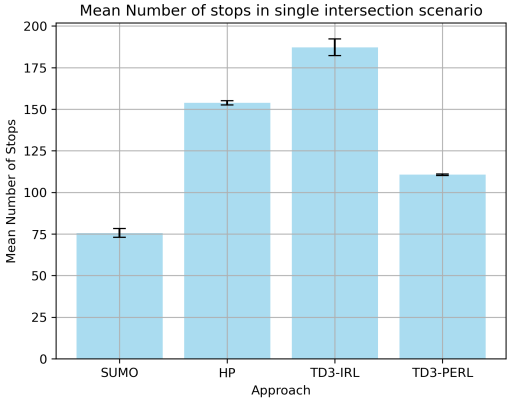

Figure 6.10: Single junction; Mean number of stops

the intersection. Note that Gini index closer to zero represents closeness to perfect equality or equalization of distribution of pheromones on all legs of the intersection.

On closer inspection, we can also note that among the other approaches, PERL seems to start decreasing at 190 steps, while the other approaches take until the 240th step to start equalizing. This indicates that PERL is $\tilde{2}0\%$ faster to tend to bring the system to equilibrium.

In Figure 6.11, we see the random initialization of pheromone concentrations on each leg of the intersection. The values are randomly sampled from a uniform distribution with mean 0 and standard deviation 1 (see Chapter 4).

As time goes on, in step 100 (Figure 6.12), you can see the behaviour of agents being influenced by the pheromone concentration in the leg. The white dots are CAV agents that are choosing to take the right of way (or go), and the black ones are CAV agents that are intending to stop or yield. Note that PERL does not contribute to consensus in agent actions.

As the pheromone changes with time (Figure 6.13), so does the behaviour of agents with it. Very similar to our previous scenario, the higher concentration of pheromone attracts agents to that zone. Agents are compelled to go (or leave) only when their zone (or leg in this case) starts to decrease in concentration relative to it.

In chapter 4, we talk about the notion of redistribution of traffic to improve flow [54]. We also examined Figure 6.9 and concluded that PERL shows a higher tendency to move to-

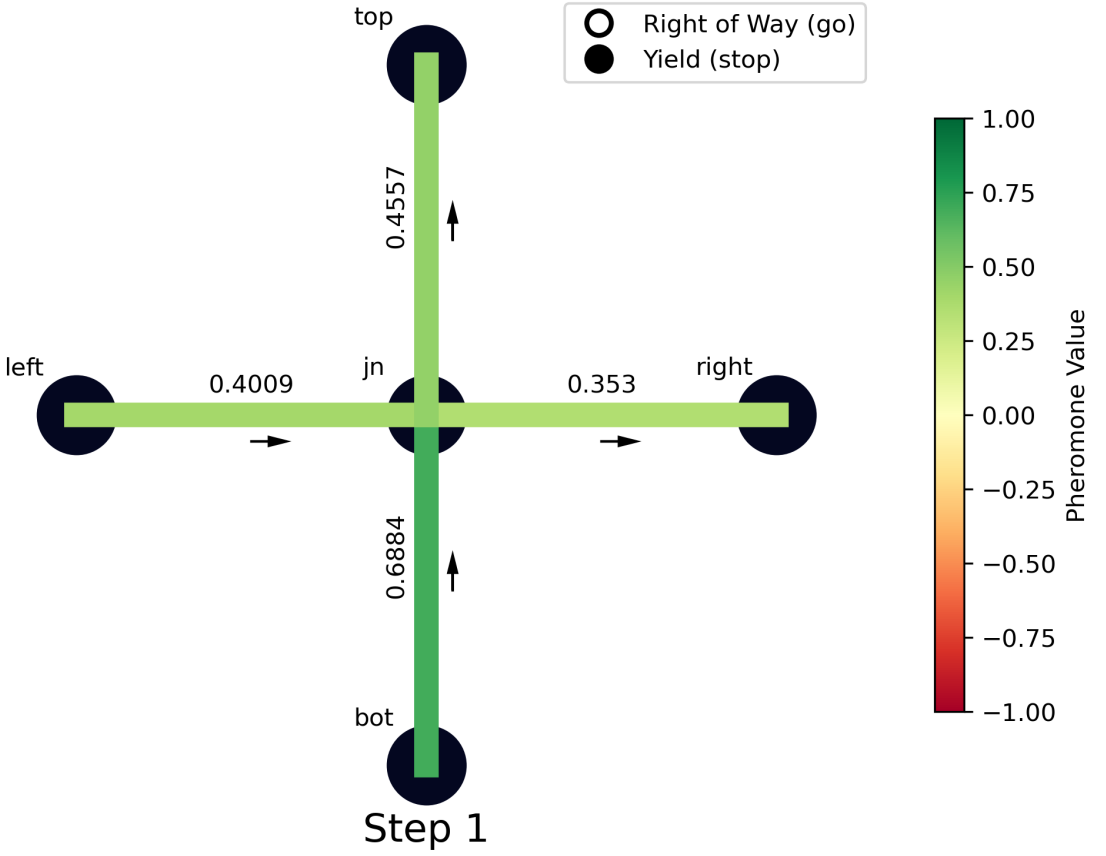

Figure 6.11: Single junction; pheromone levels and agent behaviour (step 1)

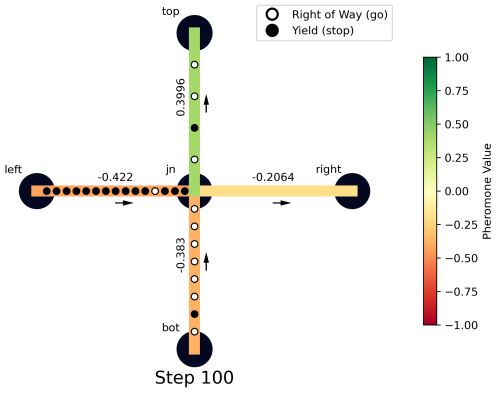

Figure 6.12: Single junction; pheromone levels and agent behaviour (step 100)

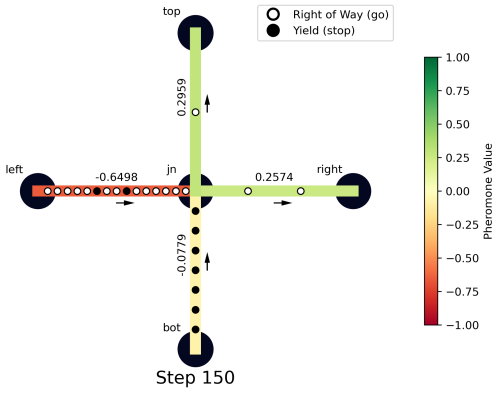

Figure 6.13: Single junction; pheromone levels and agent behaviour (step 150)

Table 6.16: Highlights of PERL vs. baselines for single intersection

Metric	**SUMO**	**HP**	**TD3-IRL**	**TD3-PERL**
Simulation time (s)	407.1 ± 1.0	371.6 ± 1.4	389.2 ± 2.33	**356.5 ± 1.0**
Mean waiting time (s)	147.25 ± 0.5	121.97 ± 1.28 ±	144.82 ± 4.98	**115.28 ± 2.67**
Mean travel time (s)	32.45 ± 0.01	37.38 ± 0.22	34.12 ± 1.84	**32.37 ± 1**
Mean pheromone distribution Gini	0.19 ± 0.02	0.18 ± 0.05	0.22 ± 0.02	**0.17 ± 0.03**
Mean pheromone distribution entropy ($bits$)	1.79 ± 0.02	1.85 ± 0.03	1.82 ± 0.06	**1.88 ± 0.02**
Mean agent distribution Gini	0.68 ± 0.01	0.56 ± 0.02	0.61 ± 0.09	**0.54 ± 0.04**
Mean agent distribution entropy ($bits$)	0.98 ± 0.01	1.05 ± 0.13	1.02 ± 0.08	**1.09 ± 0.04**

wards an equilibrium state. The mean Gini index and Shannon entropy of the methods show that PERL spent the longest time in a state closer to equilibrium in both pheromone and agent distribution. From the point of view of a single CAV agent in a partially observable non-stationary environment, the stigmergic pheromone mechanisms helped PERL to coordinate actions with other agents and guide the system to an attractor state, charactersitic of self-organization. PERL's constant objective is to move towards a state of pheromone equilibrium, resulting in a more uniform traffic distribution, ultimately improving traffic flow [54].

It follows that PERL outperforms all our baselines (see section 6.2) in simulation time taken for all vehicle agents to complete their routes, by being ∼4-13% lower than the best and worst performing among the baselines respectively. It also claims the least mean waiting time and mean travel time among the baselines. The mean waiting time is ∼6-24% (Table 6.16) lower than the best and worst performing among the baselines respectively. In particular, this experiment shows there is significance in the traffic performance metrics with respect to the pheromone and agent distribution results 6.16). This is in line with the idea that striving for a equalized distribution of traffic improves traffic flow [54].

Halting is the action or intention to stop. This does not include the time a CAV agent may spend waiting at a stopped state, after it has already come to a stop. Excessive halting can result in a break of traffic fluidity [211]. But halting too less can indicate a lack of coordination or selfish behaviour by agents as apparent in our results in Table 6.16 (Figure 6.10). Notably, PERL exhibits a relatively high number of halts when compared

to the SUMO policy, reinforcing the critical role that action step discretization plays in real-time decision-making. This discrepancy arises because MARL-based approaches like PERL execute actions at discrete time intervals (e.g., one action per second), whereas SUMO's built-in policies operate in a more continuous manner, responding to dynamic traffic conditions with finer temporal granularity. As a result, PERL agents may not always react swiftly to sudden traffic changes, leading to temporary halts as they adjust their decisions in subsequent steps. This limitation becomes particularly evident in mixed environments such as this one. Thus, while PERL demonstrates adaptive decision-making capabilities, the constraints of discrete action execution may contribute to increased instances of halting, especially in complex traffic interactions.

Additionally, it is also interesting to note that the difference in metrics like simulation time (or total system completion time) are not as exaggerated when the number of agents are lower. At a maximum of 40 agents (20 agents per leg), the system travel time was 88 for PERL compared to 84 in SUMO. This could indicate that the strengths of PERL are magnified with an increasing scale of agents.

6.5.3 Negotiation at multiple interconnected intersections

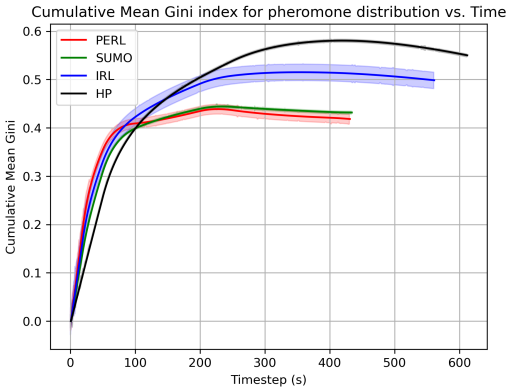

Figure 6.14: Gini index for pheromone distribution

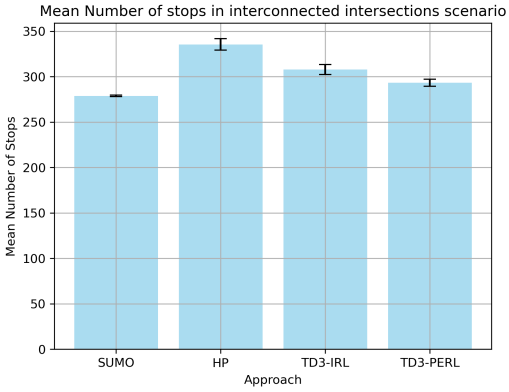

Figure 6.15: Multiple interconnected junctions; Mean number of stops

In this section, we analyze the performance metrics of multiple interconnected intersections, each with upto 100 agents for every incoming flow, totaling 400 agents.

Figure 6.14 illustrates the cumulative mean Gini index of pheromone distribution over time during evaluation. The Gini index near zero indicates a perfect equality or uniform

distribution of pheromones across all intersection legs. Figure 6.15 shows us that the mean number of stops for all methods is much higher compared to the single intersection scenario. However, the number of mean stops for PERL is higher ($\sim 5\%$) than that of SUMO as in the case of the previous scenario, continuing that trend in results.

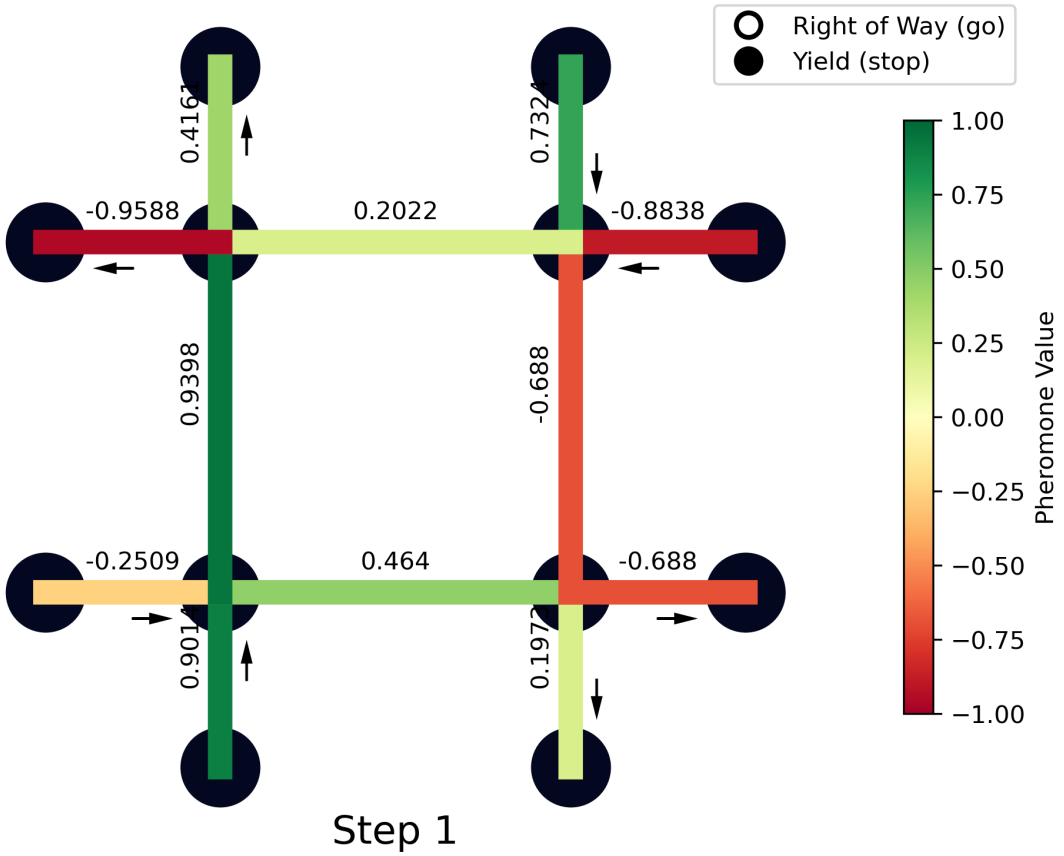

Figure 6.16: Multiple interconnected intersections; pheromone levels and agent behaviour (step 1)

Figure 6.16 depicts the random initialization of pheromone concentrations across the 12 legs of interconnected intersections. These values are randomly drawn from a uniform distribution with a mean of 0 and a standard deviation of 1 (see Chapter 4).

As time progresses, by step 50 (Figure 6.17), the influence of pheromone concentration on agent behavior becomes apparent. In the figure, white dots represent CAV agents opting to proceed, while black dots denote agents intending to yield. Notably, agents on the incoming left leg of the bottom left junction and the incoming right leg of the top right junction accumulate a large portion of negative pheromone. This accumulation and the observation that the agents are starting to convert from yielding to taking the right of way as their optimal action suggests that PERL has learned that at a certain distribution threshold of

locally perceivable pheromones, it becomes optimal to take the right of way. However, if conflicting incoming legs of both junctions fail to share the same strategy, consensus issues may arise. It's essential to emphasize that while PERL promotes coordinated emergent behavior through pheromone evaporation and diffusion processes, it does not actively contribute to consensus in agent actions.

As the pheromone levels change over time (Figure 6.18), we observe the delayed effect of the accumulation of negative repelling pheromone from step 50. It's only by step 250 that we witness the decision to take the right of way being executed with consensus among the participating CAV agents. Similar to the previous scenario, higher concentrations of pheromone attract agents to their respective zones. Agents tend to proceed or yield only when the concentration in their zone (or leg in this case) diminishes relative to others.

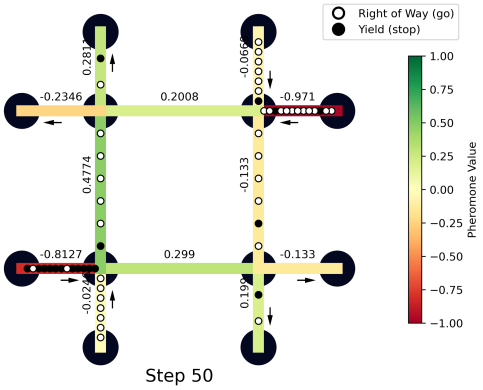

Figure 6.17: Multiple interconnected intersections; pheromone levels and agent behaviour (step 50)

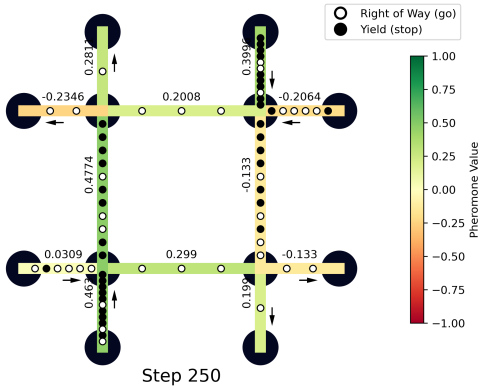

Figure 6.18: Multiple interconnected intersections; pheromone levels and agent behaviour (step 250)

Comparing the results to the single junction version, we observe less pronounced differences between PERL and the next best baseline (SUMO). This is attributed to the increased prominence of consensus among multiple agents across interconnected intersections, where PERL maintains a slight performance advantage.

We only increase the scale factor by a small margin, expanding from a single intersection to four interconnected intersections. Even with this modest increase in complexity, we observe that PERL exhibits greater variability in its results compared to SUMO's policy, as indicated by the higher standard deviation (Table 6.17) across key metrics such as simulation time, mean waiting time, and mean travel time. This inconsistency likely stems from the

absence of a direct consensus mechanism among agents; instead, PERL relies solely on the emergent effects of stigmergic processes to shape decision-making at the individual level. As the system grows in scale, the indirect nature of this coordination introduces additional uncertainty, making it more challenging to maintain stable performance. The problem is already demonstrating signs of increasing complexity, suggesting that further scaling could exacerbate these inconsistencies. Future work should explore methods to enhance PERL's scalability while preserving stability, potentially by introducing complementary mechanisms that refine coordination without undermining the decentralized nature of the approach.

Overall, PERL demonstrates better performance compared to baselines, albeit by smaller margins, even in scenarios with multiple interconnected intersections.

Table 6.17: Highlights of PERL vs. baselines for multiple interconnected intersection

Metric	**SUMO**	**HP**	**TD3-IRL**	**TD3-PERL**
Simulation time (s)	435.1 ± 0.8	613.23 ± 3.4	562.78 ± 1.93	**432.3 ± 1.0**
Mean waiting time (s)	137.26 ± 1.32	201.46 ± 3.11	209.43 ± 4.78	**136.98 ± 2.19**
Mean travel time (s)	38.66 ± 0.06	61.73 ± 0.34	49.32 ± 0.17	**37.45 ± 0.09**
Mean pheromone distribution Gini	0.44 ± 0.03	0.55 ± 0.04	0.50 ± 0.07	**0.42 ± 0.03**
Mean pheromone distribution entropy ($bits$)	2.11 ± 0.04	2.12 ± 0.03	1.78 ± 0.05	**2.23 ± 0.02**
Mean agent distribution Gini	0.62 ± 0.02	0.65 ± 0.03	0.69 ± 0.08	**0.58 ± 0.05**
Mean agent distribution entropy ($bits$)	1.59 ± 0.02	1.36 ± 0.12	1.53 ± 0.07	**1.68 ± 0.05**

6.6 Scaling Up to Real-World Environments

While this study does not use real-world network topologies in SUMO, there are several key reasons for this choice, rooted in the need for conceptual clarity, computational feasibility, and methodological rigor. Although SUMO provides real-world network topologies, using them directly is not straightforward, as they often require manual adjustments in **NetEdit** to ensure proper connectivity and functionality within the simulation.

- **Focus on Conceptual Validation:** The primary focus of this research is on the fundamental principles of self-organization and multi-agent reinforcement learning (MARL) itself. Using simplified environments allows for clearer analysis of algorithmic

behavior without interference from real-world complexities. This ensures that insights gained are directly attributable to the proposed method rather than external confounding factors.

- **Computational Constraints:** Real-world traffic networks are computationally expensive to simulate, particularly when training MARL models that require thousands of iterations. High-fidelity simulations with large-scale networks demand significant computational resources, making experimentation and iterative improvements infeasible within a reasonable timeframe. By using controlled environments, we ensure the feasibility of extensive experimentation without excessive computational overhead.
- **Controlled Experimentation:** Real-world traffic networks introduce numerous uncontrollable factors, such as irregular road layouts, traffic light configurations, weather conditions, and unexpected disruptions like road construction. These external variables make it challenging to isolate the effects of the proposed approach. By designing controlled synthetic environments, we are able to systematically test, analyze, and compare different methodologies under consistent and reproducible conditions.
- **Challenges with SUMO's Real-World Networks:** While SUMO provides real-world network topologies, these networks often require significant preprocessing before they can be used effectively in a MARL framework. Many of these networks are imported from OpenStreetMap (OSM) and other datasets, but their raw format often includes disconnected road segments, missing priority rules, or improperly defined lane connections. As a result, these networks must be manually refined in **NetEdit**, SUMO's graphical network editor, to ensure correct connectivity, realistic traffic flow, and compatibility with reinforcement learning agents.

  Given the computational cost and additional manual effort required to preprocess these networks, we opted for controlled synthetic environments that allow us to focus on the core problem of MARL-based traffic self-organization.
- **Lack of Standard Benchmarks for Comparison:** Unlike other domains in reinforcement learning, there is no universally accepted standard for evaluating MARL-based self-organization in real-world traffic networks.
- **Future Work Consideration:** While this study focuses on conceptual validation in synthetic environments, future research can explore the scalability of our approach

in real-world network topologies. One possible direction is refining and validating PERL in preprocessed SUMO networks where connectivity issues have been resolved in **NetEdit**. Extending PERL to urban mobility scenarios with real traffic data would be a logical next step in evaluating its practical applicability.

7 Conclusion and Future Work

This thesis explored the integration of stigmergic communication mechanisms, specifically Ant Colony Optimization (ACO), within Multi-Agent Reinforcement Learning (MARL) to enhance coordination in decentralized traffic systems. The research was guided by the following key question:

- **RQ:** How can stigmergic communication mechanisms, such as Ant Colony Optimization (ACO) from Swarm Intelligence (SI), enhance coordination in Multi-Agent Reinforcement Learning (MARL) for environments characterized by partial observability and non-stationarity? Furthermore, how efficiently can emergent behaviors resulting from local interactions coordinate the actions of multiple agents to achieve system-wide objectives in real-world application domains?

To address this question, we proposed and evaluated PERL, a novel MARL-based framework that incorporates digital pheromones and stigmergic processes into agent decision-making. Our hypothesis posited that this integration would induce self-organization, allowing emergent coordinated behavior to arise naturally from local interactions. By leveraging mechanisms such as pheromone evaporation, diffusion, and positive feedback, PERL aimed to guide Connected and Autonomous Vehicles (CAVs) toward more efficient traffic flow redistribution without explicit global coordination.

Findings and Contributions

The experimental results presented in this thesis provide evidence that supports the core premise of our hypothesis while also highlighting key limitations and areas for improvement. Below, we summarize the main contributions and findings:

- **C1: Demonstrating Self-Organization through Stigmergy.** The results indicate that pheromone-based reinforcement learning can induce decentralized coordination in traffic scenarios characterized by partial observability and non-stationarity. Agents successfully adapted to environmental conditions using only local stigmergic cues, validating our hypothesis that self-organization can emerge from the integration of digital pheromones with MARL.
- **C2: Performance Analysis of PERL vs. Baselines.** Compared to heuristic and MARL-based baselines, PERL demonstrated competitive performance in terms of traffic flow optimization, lane-changing efficiency, and intersection throughput. However, it exhibited higher variance in certain metrics (e.g., waiting time and recovery time), particularly in complex multi-intersection scenarios. This suggests that while stigmergy enables decentralized coordination, its stability and convergence properties require further refinement.
- **C3: Limitations and Challenges.** While PERL successfully leveraged stigmergic communication, several limitations emerged. The absence of explicit agent consensus mechanisms resulted in inconsistencies in performance across trials. Moreover, the sensitivity of the framework to initialization conditions (e.g., pheromone distribution at the start of training) introduced variability in outcomes. These findings highlight the trade-offs between flexibility and stability in purely decentralized MARL-based approaches.
- **C4: Pathways for Future Work.** To further develop PERL, future research should explore hybrid approaches that combine stigmergic cues with direct inter-agent communication. Additionally, applying PERL to real-world traffic networks, refining pheromone update rules, and investigating domain-agnostic calculations of pheromones or learned pheromones remain important directions for extending this work.

Overall, this research contributes to the field of decentralized traffic management by demonstrating how stigmergic communication can enhance coordination in MARL systems. While PERL shows promise as a self-organizing traffic control mechanism, additional work is required to ensure stability, scalability, and real-world applicability. The insights gained from this study provide a foundation for future advancements in multi-agent coordination

for intelligent transportation systems.

Impact on Swarm-Based Reinforcement Learning and Connected Autonomous Vehicles

The findings of this research have significant implications for both swarm-based reinforcement learning (RL) and the field of connected and autonomous vehicles (CAVs). By demonstrating how stigmergic communication mechanisms, such as pheromone-based signaling, can facilitate decentralized coordination in MARL, this work bridges the gap between biologically inspired swarm intelligence techniques and multi-agent decision-making in dynamic environments.

Swarm-Based RL: Traditional reinforcement learning approaches often struggle with non-stationary environments due to the shifting policies of other learning agents. By incorporating stigmergic communication, this work offers a mechanism for agents to influence their environment in a persistent yet adaptive manner. This contribution lays the groundwork for new RL paradigms where agents modify their surroundings as an implicit means of coordination, rather than relying solely on direct agent-to-agent communication. Future research in swarm-based RL can build upon this principle to develop more robust strategies for decentralized problem-solving in various domains, including robotics and distributed AI systems.

Connected and Autonomous Vehicles (CAVs): The application of stigmergy to CAV coordination presents a novel approach to addressing mixed environments like traffic. Unlike conventional centralized traffic management systems, PERL enables self-organizing traffic behavior without requiring extensive infrastructure or explicit vehicle-to-vehicle (V2V) communication. This aligns with the broader vision of scalable, decentralized traffic control systems capable of adapting to real-time conditions with minimal reliance on external coordination. However, for practical deployment, additional refinements in pheromone update mechanisms and robustness to initialization biases are necessary. Future extensions could integrate PERL into hybrid traffic control architectures that leverage both local stigmergic signals and global optimization techniques to enhance efficiency and safety

in urban mobility systems.

In summary, this research contributes to the evolution of swarm-based RL by demonstrating how biologically inspired coordination mechanisms can enhance multi-agent learning. Additionally, it opens new avenues for decentralized traffic management in CAV networks, paving the way for more adaptive and scalable solutions in intelligent transportation systems.

Adapting PERL to Other Applications and Designing the Pheromone Signaling System

While this research focuses on the application of PERL in connected and autonomous vehicle (CAV) coordination, the underlying principles of pheromone-based reinforcement learning extend to a variety of domains where decentralized coordination is essential. The stigmergic communication mechanism employed in PERL is highly adaptable and can be leveraged in other multi-agent systems, such as robotic swarms, warehouse automation, and distributed sensor networks.

Potential Applications Beyond CAVs:

- **Autonomous Drone Swarms:** In environments requiring decentralized UAV (unmanned aerial vehicle) coordination, such as search-and-rescue missions or environmental monitoring, pheromone signals could be used to mark explored regions or indicate areas of high interest. The evaporation and diffusion mechanisms would allow the swarm to efficiently allocate resources to unexplored or high-priority areas.
- **Warehouse and Factory Automation:** Autonomous robots in warehouse settings often require efficient path planning and task allocation. A pheromone-based system could guide robots toward tasks with high urgency while ensuring load balancing to prevent congestion in busy zones.
- **Internet of Things (IoT) and Smart Grid Management:** In distributed IoT systems, digital pheromones could be used for load balancing in smart grids, optimizing energy distribution based on real-time demand. Similarly, in networked sensor systems, pheromone trails could help prioritize data routing to prevent congestion in high-traffic nodes.

- **Pedestrian Flow and Crowd Management:** Pheromone-inspired signaling could be used to optimize pedestrian movement in large-scale events, such as concerts or evacuations, by dynamically guiding individuals based on congestion levels.

Designing the Pheromone Signaling System:

- **Pheromone Representation:** The pheromone should encode information relevant to decision-making. For example, in CAVs, it represents congestion levels, while in drone swarms, it might indicate search coverage or obstacle density.
- **Evaporation and Diffusion Rates:** These parameters determine the persistence and spread of pheromone signals. A high evaporation rate ensures rapid adaptability but may lead to information loss, while excessive diffusion may dilute important local information. Application-specific tuning is required to balance responsiveness and stability.
- **Agent Interaction with Pheromones:** The decision-making process should integrate pheromone signals effectively. This could be done by incorporating pheromone levels into the agent's state representation, influencing reward functions, or modifying exploration strategies. The chosen approach depends on whether the goal is reactive adaptation (e.g., obstacle avoidance) or long-term optimization (e.g., traffic flow management).

By carefully adapting these principles, PERL's stigmergic approach can be extended beyond traffic coordination to a wide range of multi-agent domains. Future work could focus on refining these adaptations and validating them in real-world testbeds across various industries.

Key Limitations of PERL

While PERL demonstrates promising results in multi-agent coordination through stigmergic reinforcement learning, there are several limitations that should be acknowledged. These limitations highlight potential areas for future research and improvements.

- **Dependence on Simulation Assumptions:** The experiments are conducted in a SUMO-based traffic simulation, which, while widely used, still abstracts many real-

world complexities such as human driving unpredictability, imperfect communication, and varying road conditions. The effectiveness of PERL in real-world deployments remains to be validated.

- **Scalability Challenges:** While PERL successfully coordinates multiple agents, its performance consistency decreases as the number of intersections increases. This suggests potential limitations in scalability when applied to large-scale urban environments without modifications to the pheromone update mechanisms.
- **Requirement of Domain Knowledge for Pheromone Calculation:** The current approach requires a domain expert to handcraft pheromone calculations to ensure they reflect specific aggregated information over time. A potential improvement could be allowing the pheromone signals to be learned dynamically from reward signals, such as for finding states not seen before, rather than being manually designed. This could make the system more adaptive and capable of generalizing to different environments. Additionally, it could be argued that PERL demands a deeper understanding of the problem domain to design a successful agent compared to traditional reinforcement learning methods. This is because, aside from standard RL components (such as state representations, reward functions, and exploration strategies), PERL also requires a well-thought-out pheromone signaling mechanism that meaningfully encodes relevant environmental information. While this additional complexity may introduce challenges in designing agents, it also provides a structured way to inject prior knowledge into the learning process. This trade-off means that although PERL may require more upfront effort in defining the pheromone system, it has the potential to achieve superior coordination and faster convergence in complex multi-agent environments where traditional RL struggles. Future work could explore ways to automate or simplify pheromone design, making PERL more accessible to a wider range of applications without requiring extensive domain expertise.
- **Lack of Direct Agent-to-Agent Communication:** PERL relies purely on indirect communication via pheromone updates, which can be beneficial for decentralized control but may limit the speed and precision of coordination in high-density traffic conditions where explicit communication (e.g., V2V messaging) could improve decision-making.

- **Fixed Hyperparameter Settings:** Although hyperparameters were chosen based on prior research, there was limited exploration of hyperparameter optimization. More adaptive approaches to adjusting parameters such as pheromone decay rates or learning rates could enhance robustness across different environments.
- **Potential Sensitivity to Initial Conditions:** The initialization of pheromone levels influences early agent behavior, which could introduce variance in system performance. A more detailed study on initialization strategies could help improve consistency in learning outcomes.

Despite these limitations, PERL provides a novel approach to integration of pheromones with MARL, laying the groundwork for future work where we could enhance its applicability to real-world autonomous systems.

Future Work and Consideration of Assumptions

While this work demonstrates the potential of integrating stigmergic principles into MARL for coordination in multi-agent systems, several assumptions were made to maintain a controlled experimental setting. Future work should explore relaxing these assumptions to better align with real-world complexities and assess the robustness of PERL in more dynamic environments. Below, we discuss each assumption and potential avenues for future research:

1. **Failure-Free Agents:** The assumption that all agents operate without failures simplifies the analysis by removing concerns related to communication breakdowns or hardware malfunctions. However, real-world autonomous systems are prone to failures, such as sensor noise, intermittent connectivity, or mechanical issues. Future work could explore how PERL adapts to agent failures by incorporating fault tolerance mechanisms, such as redundant communication channels, adaptive pheromone decay rates, or agent-based error correction strategies.
2. **Homogeneous Agents:** In this study, all agents share identical characteristics, including uniform acceleration and decision-making processes. While this assumption aids in isolating the effects of pheromone-based coordination, real-world systems involve heterogeneous agents with varying speeds, priorities, and objectives. A po-

tential research direction is extending PERL to heterogeneous environments, where agents learn to interpret pheromone signals differently based on their individual properties, leading to more adaptive and flexible coordination strategies.

3. **Synchronized System Time:** The assumption that all agents operate under synchronized time steps ensures smooth coordination and prevents conflicts in decision-making. However, real-world systems often operate asynchronously, with variable delays in sensing and actuation. Future extensions could investigate the impact of asynchronous decision-making in PERL, perhaps by introducing event-driven updates or using reinforcement learning techniques that adapt to time delays while maintaining coordination.
4. **Absence of Consensus Consideration in PERL:** PERL relies on self-organization rather than explicit consensus-building among agents. While this approach facilitates emergent behavior, certain applications may require consensus mechanisms to ensure global stability, particularly in safety-critical scenarios such as autonomous vehicle platooning or emergency traffic rerouting. A promising direction for future work is hybridizing PERL with explicit consensus methods, such as graph-based communication protocols or decentralized voting schemes, to improve stability while preserving scalability.
5. **Simple Grid-Based Networks:** This thesis evaluates PERL within structured, grid-based networks, which provide a controlled environment for studying emergent coordination. However, real-world traffic networks exhibit irregular intersections, varying road widths, and dynamic lane structures. Future research should extend PERL to more complex, real-world topologies, possibly integrating SUMO's real-world network datasets with automated connection corrections via NetEdit.

By addressing these limitations, future research can further refine PERL's applicability in large-scale, real-world environments, improving its adaptability and robustness in dynamic, uncertain settings. Expanding PERL to handle heterogeneous agents, asynchronous decision-making, and real-world road networks will be crucial for its deployment in intelligent transportation systems and beyond.

In conclusion, this thesis explores the integration of Swarm Intelligence (SI) methods

with Multi-Agent Reinforcement Learning (MARL) to address coordination challenges in partially observable and non-stationary environments such as traffic scenarios. This integration is achieved by augmenting agent states with pheromone information and rewarding states with equalized local pheromone levels. Due to the nature of diffusion, this local tendency transmits globally and incentivizes collective behavior among agents, promoting self-organization for emergent coordination to redistribute traffic, which has been shown to improve traffic flow [54]. Based on the outcomes across all scenarios, we conclude that PERL exhibits the ability to encourage coordinated emergent behavior within partially observable non-stationary environments. Additionally, our pioneering feedback signal for local pheromone distribution implicitly induces self-organization by creating a global objective through cascading [189] of feedback signals, and aids in improving traffic flow by facilitating its redistribution.

While promising results have been obtained, there remain areas for improvement and future exploration. In the real world, Connected Autonomous Vehicles (CAVs) neither emit pheromones nor allow for their storage in the environment. Thus, alternative technologies such as decentralized Vehicle-to-Vehicle (V2V) communication in VANETs and Mobile VANETs (MANETs) [59, 60] could be explored. Establishing a framework for agents to interpret the current representation of the pheromone field in their local environment is crucial for these systems. The direction of diffusion plays a pivotal role in coordinating emergent self-organization, emphasizing the need for a standardized approach if an optimal direction can be identified. Lastly, real-world traffic systems comprise heterogeneous entities, suggesting a shift from homogeneous to heterogeneous agents for further exploration.

Bibliography

[1] "Incorporating travel-time reliability into the congestion management process (cmp): A primer," 2 2015. [Online]. Available: https://ops.fhwa.dot.gov/publications/fhwahop14034/index.htm

[2] "Sae international and iso collaborate to update and refine industry-recognized sae levels of driving automation," 5 2021. [Online]. Available: https://www.sae.org/news/press-room/2021/05/sae-international-and-iso-collaborate-to-update-and-refine-industry-recognized-sae-levels-of-drivi //www.sae.org/binaries/content/assets/cm/content/blog/sae-j3016-visual-chart_5.3.21.pdf

[3] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A survey of autonomous vehicles: Enabling communication technologies and challenges," *Sensors 2021, Vol. 21, Page 706*, vol. 21, p. 706, 1 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/706/htmhttps://www.mdpi.com/1424-8220/21/3/706

[4] (2014, 4) The remarkable self-organization of ants. [Online]. Available: https://www.quantamagazine.org/ants-build-complex-structures-with-a-few-simple-rules-20140409

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction second edition*, 2018.

[6] R. Polvara, M. Patacchiola, S. Sharma, J. Wan, A. Manning, R. Sutton, and A. Cangelosi, "Autonomous quadrotor landing using deep reinforcement learning," 09 2017.

[7] A. Kalidas, C. Jackson, A. Md, S. Basheer, S. Mohan, and S. Sakri, "Deep reinforcement learning for vision-based navigation of uavs in avoiding stationary and mobile obstacles," *Drones*, vol. 7, p. 245, 04 2023.

[8] S. Liu, Z. Yang, Z. Zhang, R. Jiang, T. Ren, Y. Jiang, S. Chen, and X. Zhang, "Application of deep reinforcement learning in reconfiguration control of aircraft anti-skid braking system," *Aerospace*, vol. 9, p. 555, 09 2022.

[9] E. Bonabeau, G. Theraulaz, J. L. Deneubourg, S. Aron, and S. Camazine, "Self-organization in social insects," pp. 188–193, 1997.

[10] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," *Robots and Biological Systems: Towards a New Bionics?*, pp. 703–712, 1993. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-58069-7_38

[11] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[12] T. StÃ¼tzle and M. Dorigo, "Ant colony optimization," 01 2004.

[13] H. N. Sanghvi, "Self-organization of connected and autonomous vehicles to minimize road traffic congestion," 9 2021.

[14] O. Kilinc and G. Montana, "Multi-agent Deep Reinforcement Learning with Extremely Noisy Observations," dec 2018. [Online]. Available: https://arxiv.org/abs/1812.00922v1

[15] M. Hausknecht and P. Stone, "Deep Recurrent Q-Learning for Partially Observable MDPs," *AAAI Fall Symposium - Technical Report*, vol. FS-15-06, pp. 29–37, jul 2015. [Online]. Available: https://arxiv.org/abs/1507.06527v4

[16] J. N. Foerster, F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling, "Bayesian Action Decoder for Deep Multi-Agent Reinforcement Learning," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 3428–3442, nov 2018. [Online]. Available: https://arxiv.org/abs/1811.01458v3

[17] H. Nekoei, A. Badrinaaraayanan, A. Sinha, M. Amini, J. Rajendran, A. Mahajan, and S. Chandar, "Dealing with non-stationarity in decentralized cooperative multi-agent deep reinforcement learning via multi-timescale learning," in *Conference on Lifelong Learning Agents*. PMLR, 2023, pp. 376–398.

[18] R. R. Kumar and P. Varakantham, "On solving cooperative marl problems with a few good experiences," *arXiv preprint arXiv:2001.07993*, 2020.

[19] T. Malloy, C. R. Sims, T. Klinger, M. Liu, M. Riemer, and G. Tesauro, "Capacity-limited decentralized actor-critic for multi-agent games," *2021 IEEE Conference on Games (CoG)*, pp. 1–8, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244956107

[20] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "Metalight: Value-based meta-reinforcement learning for traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 1153–1160, 4 2020.

[21] Z. Ge, "Reinforcement learning-based signal control strategies to improve travel efficiency at urban intersection," *Proceedings - 2020 International Conference on Urban Engineering and Management Science, ICUEMS 2020*, pp. 347–351, 4 2020.

[22] S. Li, C. Wei, X. Yan, L. Ma, D. Chen, and Y. Wang, "A deep adaptive traffic signal controller with long-term planning horizon and spatial-temporal state definition under dynamic traffic fluctuations," *IEEE Access*, vol. 8, pp. 37 087–37 104, 2020.

[23] M. A. S. Kamal, T. Hayakawa, and J. I. Imura, "Development and evaluation of an adaptive traffic signal control scheme under a mixed-automated traffic scenario," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 590–602, 2 2020.

[24] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An enabling technique," *Data Min. Knowl. Discov.*, pp. 393–423, 10 2002.

[25] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability (Switzerland), Special Issue Road Traffic Engineering and Sustainable Transportation*, vol. 12, 6 2020.

[26] "Tomtom traffic survey," 2020. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking

[27] C. Systematics and T. A. T. T. Institute, "Trafï¬cc*congestionandreliability* : *Trendsandadvancedstrategiesforcongestionmitigation*,″ 2005.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature 2015 518:7540*, vol. 518, pp. 529–533, 2 2015.

[29] P. Ha-Li and D. Ke, "An intersection signal control method based on deep reinforcement learning," vol. 2017-Octob. Institute of Electrical and Electronics Engineers Inc., 10 2017, pp. 344–348.

[30] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 1243–1253, 2 2019.

[31] N. Bhave, A. Dhagavkar, K. Dhande, M. Bana, and J. Joshi, "Smart signal - adaptive traffic signal control using reinforcement learning and object detection," *Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2019*, pp. 624–628, 12 2019.

[32] K. Lin, R. Zhao, Z. Xu, D. Chuxing, and J. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, vol. 18, pp. 1774–1783, 7 2018.

[33] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," 2017.

[34] A. H. Chow, R. Sha, and Y. Li, "Adaptive control strategies for urban network traffic via a decentralized approach with user-optimal routing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 1697–1704, 4 2020.

[35] J. Lee, J. Chung, and K. Sohn, "Reinforcement learning for joint control of traffic signals in a transportation network," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 1375–1387, 2 2020.

[36] T. Chu, J. Wang, L. CodecÃ , and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 1086–1095, 3 2020.

[37] Z. Wang, "Intelligent traffic signal control based on multiple agents," *4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pp. 315–3154, 2019.

[38] T. Le, P. KovÃ¡cs, N. Walton, H. L. Vu, L. L. Andrew, and S. S. Hoogendoorn, "Decentralized signal control for urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 431–450, 9 2015.

[39] L. Adacher, A. Gemma, and G. Oliva, "Decentralized spatial decomposition for traffic signal synchronization," *Transportation Research Procedia*, vol. 3, pp. 992–1001, 1 2014.

[40] P. Zhou, X. Chen, Z. Liu, T. Braud, P. Hui, and J. Kangasharju, "Drle: Decentralized reinforcement learning at the edge for traffic light control in the iov," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 2262–2273, 4 2021.

[41] P. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "Scoot - a traffic responsive method of coordinating signals," 1981. [Online]. Available: https://trid.trb.org/view/179439

[42] C. Yu, Y. Feng, H. X. Liu, W. Ma, and X. Yang, "Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections," *Transportation Research Part B: Methodological*, vol. 112, pp. 89–112, 6 2018.

[43] B. Xu, X. J. Ban, Y. Bian, J. Wang, and K. Li, "V2i based cooperation between traffic signal and approaching automated vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 1658–1664, 7 2017.

[44] W. Sun, J. Zheng, and H. X. Liu, "A capacity maximization scheme for intersection management with automated vehicles," *Transportation Research Procedia*, vol. 23, pp. 121–136, 1 2017.

[45] X. J. Liang, S. I. Guler, and V. V. Gayah, "Signal timing optimization with connected vehicle technology: Platooning to improve computational efficiency:," *https://doi.org/10.1177/0361198118786842*, vol. 2672, pp. 81–92, 7 2018. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0361198118786842

[46] J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu, and P. Varaiya, "Platoons of connected vehicles can double throughput in urban roads," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 292–305, 4 2017.

[47] X. Qin, Y. Bian, Z. Hu, N. Sun, and M. Hu, "Distributed vehicular platoon control considering communication topology disturbances," *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020, evaluation of platoon control with communication noise and disturbances. Connections are represented by links on a DAG. The scenario chosen is closed loop. Platoon leaders are controlled by chaging speed. The controller itself is based on stabilization of Ricatti inequality. [Online]. Available: http://its.papercept.net/proceedings/ITSC20/0014.pdfhttps://ieeexplore.ieee.org/document/9294333

[48] B. Youssef and R. Hesham, "Vehicle platooning: An energy consumption perspective," *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020.

[49] M. Vallati and L. Chrpa, "A principled analysis of the interrelation between vehicular communication and reasoning capabilities of autonomous vehicles," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-November, pp. 3761–3766, 12 2018.

[50] C. M. Day, H. Li, L. M. Richardson, J. Howard, T. Platte, J. R. Sturdevant, and D. M. Bullock, "Detector-free optimization of traffic signal offsets with connected vehicle data:," *https://doi.org/10.3141/2620-06*, vol. 2620, pp. 54–68, 1 2017. [Online]. Available: https://journals.sagepub.com/doi/10.3141/2620-06

[51] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 9 2014.

[52] W. R. Ashby, V. Foerster, G. W. Zopf, and W. R. Ashby, "Principles of the self-organizing system," *E:CO Special Double Issue*, vol. 6, pp. 102–126, 2004.

[53] C. A. Holt and A. E. Roth, "The nash equilibrium: A perspective," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 3999–4002, 2004. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.0308738101

[54] J. ZiÃ^3*kowski*, *M. Zieja*, *and M. Oszczypaa*, "*Forecasting of the traffic flow distribution in the transpo*

[55] A. M. Part, R. Akcelik, J. A. Bonneson, W. Briton, K. G. Courage, R. E. DeArazoza, R. Dowling, D. B. Fambro, R. K. Giguere, F. L. Hall, D. W. Harwood, M. Kyte, J. P. Leisch, D. S. McLeod, J. Morrall, B. K. Ostrom, R. C. Pfefer, J. L. Powell, W. R. Reilly, R. P. Roess, N. M. Rouphail, R. C. Sonntag, A. Sorton, D. W. Strong, S. Teply, P.-Y. Texier, R. Troutbeck, T. I. Urbanik, J. D. Zegeer, S. A. Ardekani, G.-L. Chang, N. H. Gartner, H. C. Lieu, C. J. Messer, A. Mohaddes, K. C. Mouskos, M. Papageorgiou, A. K. Rathi, M. A. P. Taylor, and M. V. Aerde, "Transportation research record no. 1457, part 1 1994 trb distinguised lecture, part 2 traffic flow and capacity," 1994.

[56] A. Hamilton, B. Waterson, T. Cherrett, A. Robinson, and I. Snell, "The evolution of urban traffic control: changing policy and technology," *http://dx.doi.org/10.1080/03081060.2012.745318*, vol. 36, pp. 24–43, 2 2012.

[57] A. G. Sims and K. W. Dobinson, "The sydney coordinated adaptive traffic (scat) system philosophy and benefits," *IEEE Transactions on Vehicular Technology*, vol. 29, pp. 130–137, 1980.

[58] K. Pandit, D. Ghosal, H. M. Zhang, and C. N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 1459–1471, 2013.

[59] M. A. Al-shareeda, M. A. Alazzawi, M. Anbar, S. Manickam, and A. K. Al-Ani, "A comprehensive survey on vehicular ad hoc networks (vanets)," in *2021 International Conference on Advanced Computer Applications (ACA)*, 2021, pp. 156–160.

[60] D. Hetzer, M. Muehleisen, A. Kousaridas, S. Barmpounakis, S. Wendt, K. Eckert, A. Schimpe, J. Löfhede, and J. Alonso-Zarate, "5G connected and automated driving: use cases, technologies and trials in cross-border environments," *Eurasip Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–19, dec 2021. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-021-01976-6

[61] V. Gorodetskii, "Self organization and multiagent systems: I. models of multiagent self organization," *J. Comput. Syst. Sci. Int.*, pp. 256–281, 2012.

[62] G. D. M. Serugendo, M.-P. G. Irit, and A. Karageorgos, "Self-organisation and emergence in mas: An overview," *Informatica*, vol. 30, p. 45, 2006.

[63] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, pp. 115–152, 1995.

[64] D. Alves, J. V. Ast, Z. Cong, B. D. Schutter, R. BabuÅ¡ka, D. Alves, J. V. Ast, Z. Cong, B. D. Schutter, and R. BabuÅ¡ka, "Ant colony optimization for traffic dispersion routing," pp. 683–688, 2010.

[65] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, 11 2006, pheromones - weights or Q-values of actions
Components = states or possible edges. [Online]. Available: http://ieeexplore.ieee.org/document/4129846/

[66] Z. Cong, B. D. Schutter, R. BabuÅ¡ka, Z. Cong, B. D. Schutter, and R. BabuÅ¡ka, "Ant colony routing algorithm for freeway networks," *C*, vol. 37, pp. 1–19, 2013.

[67] B. Cesme and P. G. Furth, "Self-organizing traffic signals using secondary extension and dynamic coordination," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 1–15, 11 2014.

[68] C. Bernon, M. Cossentino, and J. Pavon, "An overview of current trends in european aose research," *Informatica*, vol. 29, p. 379, 2005.

[69] S. Goel, S. F. Bush, and K. Ravindranathan, "Self-organization of traffic lights for minimizing vehicle delay," *2014 International Conference on Connected Vehicles and Expo, ICCVE 2014 - Proceedings*, pp. 931–936, 2014.

[70] M. Camurri, M. Mamei, and F. Zambonelli, "Urban traffic control with co-fields," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4389 LNAI, pp. 239–253, 2006.

[71] M. Jean-Pierre, B. Christine, L. Gabriel, and G. Pierre, "Bio-inspired mechanisms for artificial self-organised systems," *Informatica*, vol. 30, p. 55, 2006.

[72] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 3*, pp. 441–461, 3 2017.

[73] A. Gupta and S. Srivastava, "Comparative analysis of ant colony and particle swarm optimization algorithms for distance optimization," *Procedia Computer Science*, vol. 173, pp. 245–253, 1 2020.

[74] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, pp. 29–41, 1996.

[75] W. D. Hamilton and E. O. Wilson, "Sociobiology. the new synthesis," *The Journal of Animal Ecology*, vol. 46, p. 975, 10 1977. [Online]. Available: https://www.jstor.org/stable/3653?origin=crossref

[76] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995. [Online]. Available: http://ieeexplore.ieee.org/document/488968/

[77] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 12 2013.

[78] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, 9 2015.

[79] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," *33rd International Conference on Machine Learning, ICML 2016*, vol. 4, pp. 2939–2947, 11 2015.

[80] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 7 2017.

[81] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, pp. 1889–1897, 2 2015.

[82] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *33rd International Conference on Machine Learning, ICML 2016*, vol. 4, pp. 2850–2869, 2 2016.

[83] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," 12 2017.

[84] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. WÃ¼nsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature 2019 575:7782*, vol. 575, pp. 350–354, 10 2019. [Online]. Available: https://www.nature.com/articles/s41586-019-1724-z

[85] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Å½Ãdek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature 2021*, pp. 1–7, 7 2021. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2

[86] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2323, 1998.

[87] 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*, vol. 2016-December, pp. 779–788, 6 2015.

[88] "Technical note q-learning," Tech. Rep., 1992.

[89] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[90] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[91] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[92] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16326763

[93] A. Baheri *et al.*, "The synergy between optimal transport theory and multi-agent reinforcement learning," *arXiv preprint arXiv:2401.10949*, 2024.

[94] S. Li, H. Chi, and T. Xie, "Multi-agent combat in non-stationary environments," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[95] P. Li, J. Hao, H. Tang, Y. Zheng, and X. Fu, "RACE: Improve multi-agent reinforcement learning with representation asymmetry and collaborative evolution," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 19 490–19 503. [Online]. Available: https://proceedings.mlr.press/v202/li23i.html

[96] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[97] J. Ruan, X. Hao, D. Li, and H. Mao, "Learning to collaborate by grouping: a consensus-oriented strategy for multi-agent reinforcement learning," *arXiv preprint arXiv:2307.15530*, 2023.

[98] G. Jing, H. Bai, J. George, A. Chakrabortty, and P. K. Sharma, "Distributed cooperative multi-agent reinforcement learning with directed coordination graph," in *2022 American Control Conference (ACC)*, 2022, pp. 3273–3278.

[99] V. Guleva, E. Shikov, K. Bochenina, S. Kovalchuk, A. Alodjants, and A. Boukhanovsky, "Emerging complexity in distributed intelligent systems," *Entropy*, vol. 22, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/12/1437

[100] S. Bhalla, S. Ganapathi Subramanian, and M. Crowley, "Deep multi agent reinforcement learning for autonomous driving," in *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13â15, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 67â78. [Online]. Available: https://doi.org/10.1007/978-3-030-47358-7_7

[101] J. Yu, P.-A. Laharotte, Y. Han, and L. Leclercq, "Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 154, p. 104281, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X2300270X

[102] S. Liu, W. Liu, W. Chen, G. Tian, and Y. Liu, "Multi-agent cooperation via unsupervised learning of joint intentions," *arXiv preprint arXiv:2307.02200*, 2023.

[103] W. Khlifi, S. Singh, O. Mahjoub, R. de Kock, A. Vall, R. Gorsane, and A. Pretorius, "On diagnostics for understanding agent training behaviour in cooperative marl," *arXiv preprint arXiv:2312.08468*, 2023.

[104] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[105] "Research on queue equilibrium control algorithm of urban traffic based on game theory," 2023.

[106] "Game theory-based ramp merging for mixed traffic with unity-sumo co-simulation," 2022.

[107] "Equilibrium modeling of mixed autonomy traffic flow based on game theory," 2022.

[108] "Quantum game theory approach for data network routing: a solution for the congestion problem," 2022.

[109] "Mitigation of routing congestion on data networks: A quantum game theory approach," 2022.

[110] *A Game Theory-based Mechanism to Optimize the Traffic Congestion in VANETs*, 2020.

[111] "A traffic congestion analysis by user equilibrium and system optimum with incomplete information," 2020.

[112] *Optimization of Traffic Signal Control with Different Game Theoretical Strategies*, 2019.

[113] *Optimization of Traffic Signal Control Based on Game Theoretical Framework*, 2019.

[114] "Smart fuzzy logic-based model of traffic light management," 2023.

[115] *Density Based Fuzzy Logic Control Scheme for a Multi-Stoplight Road Network Implemented on a Simplified Traffic Model*, 2022.

[116] "Improving the road and traffic control prediction based on fuzzy logic approach in multiple intersections," 2022.

[117] Y. Han, H. Lee, and Y. Kim, "Extensible prototype learning for real-time traffic signal control," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 9, pp. 1181–1198, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12955

[118] "Model-based graph reinforcement learning for inductive traffic signal control," *arXiv.org*, 2022.

[119] S. Shen, G. Shen, Y. Shen, D. Liu, X. Yang, and X. K. and, "Pga: An efficient adaptive traffic signal timing optimization scheme using actor-critic reinforcement learning algorithm," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 11, pp. 4268–4289, November 2020.

[120] *PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network*, 2019.

[121] "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," vol. 34, no. 04, pp. 3414–3421, 2020.

[122] F. Mao, Z. Li, Y. Lin, and L. Li, "Mastering arterial traffic signal control with multi-agent attention-based soft actor-critic model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3129–3144, March 2023.

[123] L. Li, R. Li, Y. Peng, C. Huang, and J. Yuan, "Cooperative max-pressure enhanced traffic signal control," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 4173â4177. [Online]. Available: https://doi.org/10.1145/3511808.3557569

[124] S. Mariani and F. Zambonelli, "Learning Stigmergic Communication for Self-organising Coordination," *Proceedings - 2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2023*, pp. 47–56, 2023.

[125] Z. Cao, M. Shi, Z. Zhao, and X. Ma, "PooL: Pheromone-inspired Communication Framework for Large Scale Multi-Agent Reinforcement Learning," *arXiv*, feb 2022. [Online]. Available: https://arxiv.org/abs/2202.09722v1

[126] S. Na, H. Niu, B. Lennox, and F. Arvin, "Universal Artificial Pheromone Framework with Deep Reinforcement Learning for Robotic Systems," *2021 6th International Conference on Control and Robotics Engineering, ICCRE 2021*, pp. 28–32, apr 2021.

[127] T. H. Nguyen and J. J. Jung, "Swarm intelligence-based green optimization framework for sustainable transportation," *Sustainable Cities and Society*, vol. 71, 2021.

[128] K. Zhang, Y. Hou, H. Yu, W. Zhu, L. Feng, and Q. Zhang, "Pheromone Based Independent Reinforcement Learning for Multiagent Navigation," in *Communications in Computer and Information Science*. Springer Science and Business Media Deutschland GmbH, 2021, vol. 1449, pp. 44–58. [Online]. Available: https://link.springer.com/10.1007/978-981-16-5188-5_4

[129] A. A. Nguyen, "Scalable, Decentralized Multi-Agent Reinforcement Learning Methods Inspired by Stigmergy and Ant Colonies," Tech. Rep., 2021.

[130] T.-H. Nguyen and J. J. Jung, "Ant colony optimization-based traffic routing with intersection negotiation for connected vehicles," *Applied Soft Computing*, vol. 112, p. 107828, nov 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494621007493

[131] S. Sreenivasamurthy and K. Obraczka, "Towards Biologically Inspired Decentralized Platooning for Autonomous Vehicles," *IEEE Vehicular Technology Conference*, vol. 2021-April, apr 2021.

[132] T.-H. Nguyen and J. J. Jung, "Inverse Pheromone-based Decen-tralized Route Guidance for Connected Vehicles," *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, vol. 5, no. 21, 2021. [Online]. Available: https://doi.org/10.1145/3412841.3441925

[133] X. Xu, R. Li, Z. Zhao, and H. Zhang, "Stigmergic Independent Reinforcement Learning for Multiagent Collaboration," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[134] N. Monekosso and P. Remagnino, "Phe-q : A pheromone based q-learning," in *AI 2001: Advances in Artificial Intelligence*, M. Stumptner, D. Corbett, and M. Brooks, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 345–355.

[135] N. Monekosso, P. Remagnino, and A. Szarowicz, "An improved q-learning algorithm using synthetic pheromones," in *From Theory to Practice in Multi-Agent Systems*, B. Dunin-Keplicz and E. Nawarecki, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 197–206.

[136] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, R. Delle Monache, Maria Laura andcruise$_c$*ontrol*2 *Lysecky*, *B. Seibold*, *J. Sprinkleet al.*, "*Arecommerciallyimplementedadaptivecruis* $-7003, 2020.

[137] G. Gunter, C. Janssen, W. Barbour, R. Stern, and D. Work, "Model-based string stability of adaptive cruise control systems using field data," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 90–99, Mar. 2020, publisher Copyright: © 2016 IEEE.

[138] A. Taylor and A. Ames, "Adaptive safety with control barrier functions," 07 2020.

[139] B. Gao, K. Cai, T. Qu, Y. Hu, and H. Chen, "Personalized adaptive cruise control based on online driving style recognition technology and model predictive control," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–1, 08 2020.

[140] M. Makridis, K. Mattas, and B. Ciuffo, "Response time and time headway of an adaptive cruise control. an empirical characterization and potential impacts on road capacity," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–10, 10 2019.

[141] Y. Zhu, D. Zhao, and H. He, "Synthesis of cooperative adaptive cruise control with feedforward strategies," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–1, 02 2020.

[142] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," *arXiv preprint arXiv:2004.01339*, 2020.

[143] Y. Lin, J. McPhee, and N. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. PP, pp. 1–1, 07 2020.

[144] H. Wen, H. Li, Z. Wang, X. Hou, and K. He, "Application of ddpg-based collision avoidance algorithm in air traffic control," 12 2019, pp. 130–133.

[145] A. Rodionova, Y. Pant, K. Jang, H. Abbas, and R. Mangharam, "Learning-to-fly: Learning-based collision avoidance for scalable urban air mobility," 09 2020, pp. 1–8.

[146] T. Peng, L. Su, R. Zhang, Z. Guan, H. Zhao, Z. Qiu, C. Zong, and H. Xu, "A new safe lane-change trajectory model and collision avoidance control method for automatic driving vehicles," *Expert Systems with Applications*, vol. 141, p. 112953, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419306712

[147] L. Schester and L. E. Ortiz, "Automated driving highway traffic merging using deep multi-agent reinforcement learning in continuous state-action spaces," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 280–287.

[148] D. Pandya and H. Mer, *Inter-Vehicular Communication for Intelligent Collision Avoidance Using Machine Learning: An Overview*, 06 2021, pp. 151–159.

[149] Y. A. Ahmed, M. A. Hannan, M. Y. Oraby, and A. Maimun, "Colregs compliant fuzzy-based collision avoidance system for multiple ship encounters," *Journal of Marine Science and Engineering*, vol. 9, no. 8, 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/8/790

[150] Y. Yuan, R. Tasik, S. S. Adhatarao, Y. Yuan, Z. Liu, and X. Fu, "RACE: Reinforced Cooperative Autonomous Vehicle Collision AvoidancE," 2020.

[151] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 433–443, 1 2020.

[152] G. Bacchiani, D. Molinari, and M. Patander, "Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning," *AAMAS*, 3 2019.

[153] S. Na, H. Niu, B. Lennox, and F. Arvin, "Bio-inspired collision avoidance in swarm systems via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2511–2526, Jan. 2022.

[154] X. Zhang, X. Gao, K. Wu, and Z. Pan, "Learning neural traffic rules," *IEEE Robotics and Automation Letters*, 2024.

[155] M. Ghimire, M. Roy, and G. Lagudu, "Lane change decision-making through deep reinforcement learning," 12 2021.

[156] J. Yang, A. Nakhaei, D. Isele, K. Fujimura, and H. Zha, "Cm3: Cooperative multi-goal multi-stage multi-agent reinforcement learning," *arXiv preprint arXiv:1809.05188*, 2018.

[157] G. Wang, J. Hu, Z. Li, and L. Li, "Harmonious lane changing via deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–9, 01 2021.

[158] Z. Liang, J. Cao, S. Jiang, D. Saxena, and H. Xu, "Hierarchical reinforcement learning with opponent modeling for distributed multi-agent cooperation," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 884–894.

[159] J. Zhang, C. Chang, X. Zeng, and L. Li, "Multi-agent drl-based lane change with right-of-way collaboration awareness," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, 10 2022.

[160] L. C. Das and M. Won, "Lcs-tf: Multi-agent deep reinforcement learning-based intelligent lane-change system for improving traffic flow," *arXiv preprint arXiv:2303.09070*, 2023.

[161] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, and W. Ge, "Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic," *Autonomous Intelligent Systems*, vol. 2, no. 1, pp. 1–11, dec 2022. [Online]. Available: https://link.springer.com/article/10.1007/s43684-022-00023-5

[162] K. Sun, X. Zhao, and X. Wu, "A cooperative lane change model for connected and autonomous vehicles on two lanes highway by considering the traffic efficiency on both lanes," *Transportation Research Interdisciplinary Perspectives*, vol. 9, p. 100310, mar 2021.

[163] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, and G. Tan, "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 2*, pp. 735–748, 2 2020.

[164] J. Park, K. Min, and K. Huh, "Multi-agent deep reinforcement learning for cooperative driving in crowded traffic scenarios," 2019, pp. 1–2.

[165] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, and W. Ge, "Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 5, 2022.

[166] C. Killing, A. Villaflor, and J. M. Dolan, "Learning to robustly negotiate bi-directional lane usage in high-conflict driving scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8090–8096.

[167] M. Burov, N. Mehr, S. Smith, A. Kurzhanskiy, and M. Arcak, "Platoon formation algorithm for minimizing travel time," *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020.

[168] M. Mynuddin, W. Gao, and Z. P. Jiang, "Reinforcement learning for multi-agent systems with an application to distributed predictive cruise control," vol. 2020-July. Institute of Electrical and Electronics Engineers Inc., 7 2020, pp. 315–320.

[169] A. Peake, J. McCalmon, B. Raiford, T. Liu, S. Alqahtani, A. Peake, J. McCalmon, B. Raiford, T. Liu, and S. Alqahtani, "Multi-agent reinforcement learning for cooperative adaptive cruise control," vol. 2020-Novem. IEEE Computer Society, 11 2020, pp. 15–22.

[170] M. Guo, P. Wang, C.-Y. Chan, and S. Askary, "A reinforcement learning approach for intelligent traffic signal control at urban intersections," 10 2019, pp. 4242–4247.

[171] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[172] Y. Chin, W. Kow, W. Khong, M. K. Tan, and K. Teo, "Q-learning traffic signal optimization within multiple intersections traffic network," 11 2012, pp. 343–348.

[173] H. Joo, S. Ahmed, and Y. Lim, "Traffic signal control for smart cities using reinforcement learning," *Computer Communications*, vol. 154, 03 2020.

[174] F. Mao, Z. Li, Y. Lin, and L. Li, "Mastering Arterial Traffic Signal Control With Multi-Agent Attention-Based Soft Actor-Critic Model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3129–3144, mar 2023.

[175] C. R. Tang, J. W. Hsieh, and S. Y. Teng, "Cooperative Multi-Objective Reinforcement Learning for Traffic Signal Control and Carbon Emission Reduction," jun 2023. [Online]. Available: https://arxiv.org/abs/2306.09662v2

[176] H. Goel, Y. Zhang, M. Damani, and G. Sartoretti, "SocialLight: Distributed Cooperation Learning towards Network-Wide Traffic Signal Control," *Adaptive Agents and Multi-Agent Systems*, 2023.

[177] L. Li, R. Li, Y. Peng, C. Huang, and J. Yuan, "Cooperative Max-Pressure Enhanced Traffic Signal Control," *International Conference on Information and Knowledge Management, Proceedings*, pp. 4173–4177, oct 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3511808.3557569

[178] X. Du, J. Wang, S. Chen, and Z. Liu, "Multi-agent Deep Reinforcement Learning with Spatio-Temporal Feature Fusion for Traffic Signal Control," *ECML/PKDD*, vol. 12978 LNAI, pp. 470–485, 2021. [Online]. Available: https://2021.ecmlpkdd.org/wp-content/uploads/2021/07/sub_593.pdf

[179] W. Qu, J. Li, L. Yang, D. Li, S. Liu, Q. Zhao, and Y. Qi, "Short-term intersection traffic flow forecasting," *Sustainability*, vol. 12, p. 8158, 10 2020.

[180] U. Gunarathna, S. Karunasekara, R. Borovica-Gajic, and E. Tanin, "Intelligent autonomous intersection management," *arXiv preprint arXiv:2202.04224*, 2022.

[181] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence," oct 1999. [Online]. Available: https://oxford.universitypressscholarship.com/view/10.1093/oso/9780195131581.001.0001/isbn-9780195131581

[182] S. Na, Y. Qiu, A. E. Turgut, J. Ulrich, T. KrajnÃk, S. Yue, B. Lennox, and F. Arvin, "Bio-inspired artificial pheromone system for swarm robotics applications," *Adaptive Behavior*, vol. 29, no. 4, pp. 395–415, 2021. [Online]. Available: https://doi.org/10.1177/1059712320918936

[183] C. Bishop, *Pattern Recognition and Machine Learning*, 01 2006, vol. 16, pp. 140–155.

[184] F. W. Vook, A. Ghosh, E. Diarte, and M. Murphy, "5g new radio: Overview and performance," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1247–1251.

[185] S. Yitzhaki, "Gini's mean difference: A superior measure of variability for non-normal distributions," *Metron - International Journal of Statistics*, vol. LXI, pp. 285–316, 02 2003.

[186] C. Fowdur, "Improving traffic fluidity and road safety using cellular automata," *International Journal of Traffic and Transportation Engineering*, 12 2014.

[187] K. Jajuga and M. Walesiak, *Standardisation of Data Set under Different Measurement Scales*, 01 2000, pp. 105–112.

[188] P. Muhammad Ali and R. Faraj, "Data normalization and standardization: A technical report," Tech. Rep., 01 2014.

[189] R. M. MAY, "Will a large complex system be stable?" *Nature*, vol. 238, no. 5364, pp. 413–414, Aug 1972. [Online]. Available: https://doi.org/10.1038/238413a0

[190] H. Lu, D. Herman, and Y. Yu, "Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=TjEzIsyEsQ6

[191] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, p. 26, 2022.

[192] C. Y. Kaya and H. Maurer, "Optimization over the pareto front of nonconvex multi-objective optimal control problems," *Computational Optimization and Applications*, vol. 86, no. 3, pp. 1247–1274, 2023.

[193] H. Van Dyke Parunak, S. A. Brueckner, and J. Sauter, "Digital pheromones for coordination of unmanned vehicles," in *Environments for Multi-Agent Systems*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 246–263.

[194] Sumo api. SUMO. [Online]. Available: https://sumo.dlr.de/docs/Libsumo.html

[195] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," 2018.

[196] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[197] J. K. Terry, B. Black, A. Hari, L. S. Santos, C. Dieffendahl, N. L. Williams, Y. Lokesh, C. Horsch, and P. Ravi, "Pettingzoo: Gym for multi-agent reinforcement learning," *CoRR*, vol. abs/2009.14471, 2020. [Online]. Available: https://arxiv.org/abs/2009.14471

[198] Yaml. YAML. [Online]. Available: https://yaml.org/

[199] "Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics."

[200] J. Hare, "Dealing with sparse rewards in reinforcement learning," *arXiv preprint arXiv:1910.09281*, 2019.

[201] J. Erdmann, "Lane-changing model in sumo," vol. 24, 05 2014.

[202] (2024) Junction model parameters. [Online]. Available: https://sumo.dlr.de/docs/Simulation/Intersections.html

[203] (2024) Lane-changing model parameters. [Online]. Available: https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html#lane-changing_models

[204] L. Ceriani and P. Verme, "The origins of the gini index: extracts from variabilita e mutabilita (1912) by corrado gini," *Journal of Economic Inequality - J ECON INEQUAL*, vol. 10, pp. 1–23, 09 2012.

[205] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[206] M. O. Lorenz, "Methods of measuring the concentration of wealth," *Publications of the American Statistical Association*, vol. 9, no. 70, pp. 209–219, 1905. [Online]. Available: http://www.jstor.org/stable/2276207

[207] A. Arora, C. Meister, and R. Cotterell, "Estimating the entropy of linguistic distributions," *arXiv preprint arXiv:2204.01469*, 2022.

[208] E. H. Lieb and J. Yngvason, *A Guide to Entropy and the Second Law of Thermodynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 353–363. [Online]. Available: https://doi.org/10.1007/978-3-662-10018-9_19

[209] M. Zampieri, "Reconciling the ecological and engineering definitions of resilience," *Ecosphere*, vol. 12, 02 2021.

[210] C. Ma and D. Li, "A review of vehicle lane change research," *Physica A: Statistical Mechanics and its Applications*, vol. 626, p. 129060, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437123006155

[211] J. E. Park, W. Byun, Y. Kim, H. Ahn, and D. K. Shin, "The impact of automated vehicles on traffic flow and road capacity on urban road networks," *Journal of Advanced Transportation*, vol. 2021, p. 8404951, Nov 2021. [Online]. Available: https://doi.org/10.1155/2021/8404951